OPERATIONAL AND TACTICAL STRATEGIES FOR MANAGING SERVICE NETWORKS

A Dissertation
Presented to
The Academic Faculty

By

Yassine Ridouane

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology

December 2020

OPERATIONAL AND TACTICAL STRATEGIES FOR MANAGING SERVICE NETWORKS

Approved by:

Dr. Martin W.P. Savelsbergh, Advisor
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Natashia Boland, Co-advisor
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Alan Erera, Co-advisor
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Pascal Van Hentenryck
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Sushil Poudel
Supply Chain Solutions
United Parcel Service

Date Approved: November 30, 2020

Knowledge without action is wastefulness and action without knowledge is foolishness

Abu Hamid Al-Ghazali

To my grandmother Tudah.

ACKNOWLEDGEMENTS

First and foremost, I am grateful to God for giving me the health, the patience and the determination in this journey towards the completion of my doctoral thesis. “And say: My Lord! Increase me in knowledge”.

I dedicate this work to my family, to my village Ait Aissa Ouaali, and to my country Morocco. I am very grateful to my parents for their unconditional love and support. My mother Biya, a woman who never had the chance to sit on a school bench, was always a source of motivation for me. I will never forget all her sacrifices that made it possible for me to study in great conditions. My father Mohamed instilled in me the avid desire to pursue and excel in studies. He always urged me to think big, aim high, and never be satisfied with small achievements. I also dedicate this work to my grandmother Tudah and my aunt Khadija who left this world in the middle of the PhD program. My grandmother was such a wise and strong woman who raised an entire family of five children as a widow at a young age despite all the difficulties of life in her time. This thesis is also a dedication to my dear siblings Mohammed and Noura who are always there for me.

As I defended my thesis in the middle of the coronavirus pandemic, I want to dedicate this work to many family members who deceased, including a dear family friend in the position of a father to me, Haj Sbaai, who succumbed to the virus about two days before my defense.

This thesis would not materialize without the support and guidance of my advisors Martin Savelsbergh, Natashia Boland and Alan Erera. I am very indebted to each one of them. I am very honored and fortunate to work under the guidance of Martin whom I learned a lot from and who advised me in every step of this work. Martin had his signature in every contribution of this thesis. I am grateful to Natashia who introduced me to research in Mixed Integer Programming during my Master’s degree and encouraged me to apply for the PhD program. I am very thankful to Alan for his critical remarks and comments which shaped several aspects of this thesis. I am also very grateful to Benjamin Haaland who supported me tremendously during my Master’s and PhD

Program and helped me publish my first paper ever in the field of Statistics. I am very thankful to all the faculty body for the excellence in their teaching and mentorship, and the administrative staff of ISyE, especially Amanda Ford and Dima Nazzal for their support and help.

The content of this thesis is the fruit of a collaboration with the UPS Transportation Analytics and Operations Research team. I am very grateful to Sushil Poudel, Lianhua Long, Jack McPherson, Negin Ebadi, Michael Downey and others who helped me a lot in understanding the challenges in their business and devising the right methodology to address them. Thanks to them, I was able to have access to valuable data and generate many realistic instances to test our proposed methodology. I am also very grateful to my lab mates Ian Herszterg, Yu Yang and Ritesh Ojha whom I collaborated with in this work. I acquired many skills and learned a lot from each one of them.

During my studies at Georgia Institute of Technology, I met many wonderful friends who left a great impact on me. I would like to thank Harold, Reem, Daniela, Fabien, Seyma, Damian, Tony, Timur, Mina, Asteroid, Luke and so many others.

I would like to thank my roommates, who are like brothers to me, Ishaque, Taofiq, Mahmoud and Ali. These people supported me a lot in moments of happiness and difficulty. I also would like to thank my great friends Hasan, Shabab, Muzzamil, Jawad, Zeshan, Mohamed and his brother Hicham, Hamza, Clay and so many others. These people made my stay very pleasant in Atlanta.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis focuses on two independent aspects of service network planning. The first aspect is operational and is related to load plan adjustment in Less-Than-Truckload (LTL) freight networks. The second one is both tactical and operational and is related to equipment management in small package networks. In the first part, we suggest near real-time routing and load plan adjustment strategies to improve the system-wide daily performance of the freight network. In the second part, we present different strategies for managing inventory levels of different equipment types in a large-scale network.

We start by designing and implementing decision support technology to assist dispatchers in the daily management of load plans in LTL networks. The freight volume that enters a service network on the day of operations deviates from the forecast freight volume used to create the load plan. These deviations cause inefficiencies when the capacity on planned freight paths is no longer sufficient and delays result in missed service promises. Near real-time load plan adjustments, i.e., rerouting freight on alternate paths, can improve on-time performance without incurring additional cost (e.g., without purchasing additional capacity). We model the problem of identifying effective alternate freight paths on a time-expanded network and we develop fast heuristics for its solution so as to ensure that there is sufficient time to put the adjusted load plan in place. The load plan adjustment technology has been extensively tested using data from a large US LTL carrier. The results show that on-time performance can be improved without increasing cost, i.e., by rerouting freight and using existing capacity in the service network.

Next, we develop efficient and effective short-term equipment management strategies for small package express carriers. We start by investigating substitution-based equipment balancing for carriers operating multiple equipment types in their service network. The weekly schedule of movements used to transport packages through the service network leads to changes in equipment

inventory at the facilities in the network. We seek to reduce this change, i.e., the equipment imbalance associated with the schedule of movements, by substituting the equipment types initially assigned to the movements. We model this problem using a hierarchical optimization approach and suggest two heuristics to solve it. We also explore the value of integrating empty repositioning decisions in the model. Furthermore, we performed a computational study using real-world instances to analyze the performance of an integer programming based solution approaches and assess the benefits of substitution-based equipment balancing and integrating empty repositioning.

Finally, we shift from the previous equipment balancing perspective to an inventory aware equipment management perspective where the time dimension is considered. We formulate a mixed integer program (MIP) that tracks the inventory of each equipment type at each facility and seeks to minimize the cost of empty repositioning required to execute a given load plan and prevent stock-out occurrences, by substituting the equipment type assigned to loaded moves (respecting compatibility requirements) and adding new empty movements between facilities. We analyze the complexity of some special settings of the problem, propose a parsimonious time-discretization to control the size of the model, and introduce a dynamic variable generation algorithm to solve it. Computational experiments show that a significant reduction in the cost of empty movements required in the network can be obtained and using appropriately chosen equipment substitutions.

CHAPTER 1

INTRODUCTION AND BACKGROUND

Ground transportation forms the backbone of many economies as it can cost-effectively connect dispersed supply and demand. In the United States, trucking industry represents the main segment in the ground logistics thanks to the high density and connectivity of the existing road and rail networks. The major players in ground transportation (both small package and freight delivery) operate large service networks. For instance, the UPS small package network has more than 1,800 operating facilities where parcels are processed, and operates a multi-type delivery fleet comprised of 125 thousand vehicles. It also operates a freight consolidation network with more than 200 service centers and 22 thousand trailers[1]. Similarly, FedEx Ground has more than 600 operating facilities and operates more than 70 thousand motorized vehicles. It also operates a freight network with about 370 service centers and 25 thousand trailers[2]. With the surge of e-commerce transactions, the service networks of these carriers are expected to grow in scale as the demand for parcel delivery and freight is constantly increasing. The penetration of the internet and the shift of customer shopping trends towards online marketplaces contribute significantly to this increase and adds more challenges to the trucking industry. Based on a study [1] by the American Transportation Research Institute (ATRI) about the impact of e-commerce growth on the trucking industry, it states that e-commerce is disrupting the retail and logistics business models as it pushes towards omni-channel retailing, and thus brings many opportunities and challenges for the different stakeholders. This requires trucking companies to be flexible and adapt to these changes so as to benefit from this source of growing demand for truck transportation. Planning for these large service networks naturally comes with a multitude of challenges as there are multiple interwoven

[1] UPS Fact Sheet 2020

[2] FedEx Fact Sheet 2020

aspects to it such as the classical network design, resource and fleet planning, route planning, etc. These aspects have been historically well explored and studied in literature. Even more, operating these service networks on a daily basis brings another layer of complexity as the decision time is limited and it requires to hedge against uncertainty in the demand by constantly monitoring the network status to account for new information in a real-time or near real-time setting.

Some of the e-commerce players, such as Amazon.com, have started investing in their own multi-modal package delivery capability, which spans first mile, middle mile, and last mile logistics. This strategic step towards in-house shipping enables these companies to optimize their supply chain from supply sources to fulfillment centers and from fulfillment centers all the way to the doorsteps of their customers. It also reduces their reliance on third party logistics companies and gives more control over their shipping expenditure. This development pushes the existing logistics companies to invest in improving their operational efficiency and reduce their transportation costs so as to remain competitive and not lose their market shares. Achieving operational efficiency includes, among others, investing in technology such as smart hubs with highly automated sortation technology, better demand forecasting, and agile and robust planning strategies for both day-to-day operations and longer term plans. The focus of the research presented in this thesis tackles the latter. Our objective is to present both operational and tactical strategies that help service networks achieve efficiency through agility in planning for both small package and freight service networks.

There are many challenges that logistics companies face to which we strive to bring solutions. One important challenge is how to prepare for future demand that is characterized by high fluctuations. Demand forecasting for small package and freight, although historically widely addressed in literature (e.g., [2], [3], and [4]), remains a nontrivial task for companies. There are always rare, unforeseen events that significantly disrupt demand patterns, and most of the times, companies are not prepared for it. For instance, in the middle of the COVID-19 pandemic, logistics companies were experiencing a surge and high fluctuations in the demand as there was a sudden

shift to on-line grocery shopping, deemed as a safer option to in-store shopping. Although, this surge in the demand was a positive externality to logistics companies as it increased their revenue from e-commerce, it also increased their operational expenditure as their networks didn't have the capability to handle large volumes of non-anticipated demand that is constantly changing. For instance, in its last quarterly report in 2019, Amazon saw a 49 percent increase in delivery costs and a 30 percent drop in profit despite the surge in the delivered volume[3]. Even in normal conditions outside special events such as a pandemic, demand remains uncertain and its patterns can vary from day to day, creating discrepancy in comparison to the available capacity planned ahead of time based on an average day forecast. This uncertainty in the demand can be pronounced for new companies with recently established logistics capability such as Amazon.com that has seen a staggering growth in the delivery volumes and had to increase their network size in a short amount of time to meet the demand that can drastically change from one week to another. Based on a report by Morgan Stanley[4], Amazon.com is poised to surpass companies like UPS and Fedex in terms of the yearly delivered volume given a compound annual growth rate of 68% from 2018 to 2022. It will also expected to deliver 65% of its own e-commerce orders and 35% of third party e-commerce orders by 2022. This also affects other more established logistics companies as their share of e-commerce order delivery became significant, and therefore, are expected to be ready to react to any sudden surge in demand to maintain customer service guarantees. This daily variation requires companies to adjust their plans in near-real time setting to account for new information about the state of the network. This information may include new shipment volumes that enter the network, delays in trailer departures or arrivals, ah-hoc loads that were planned locally by terminal managers, or cancellations of existing loads, etc. Reacting in a timely manner to this new information is crucial for these logistics companies as it enables them to control their operational costs by (a) anticipating any need for re-routing shipments through alternative routes with

[3]https://info.transportationinsight.com/blog/ecommerce-delivery

[4]https://www.supplychaindive.com/news/amazon-logistics-volume-surpass-ups-fedex-2022-morgan-stanley/569044/

the existing capacity, (b) adjusting the latter by canceling any excess of capacity in some lanes and judiciously purchasing last minute extra capacity in other lanes, and (c) consolidating some loads and allowing direct movements to farther destinations to avoid any unnecessary processing at consolidations points. Some of those operations can be expensive on the day of operations such as tendering some new loads to individual contractors or third party logistics companies considering the small lead time. The ability to reach agility in day-to-day planning can bring a lot of savings in operational expenditure for logistics companies. In a recent report by the Boston Consulting Group (BCG)[5] related to the corona pandemic, it is highlighted that "the ability to react quickly to changed circumstances will have a major impact on companies' ability to contain the damage and make the most of new opportunities". This can be generalized to other situations when the competition becomes fierce between players. In this case, cost, speed, and resilience are the key factors to optimize for so as to remain competitive.

This operational aspect, i.e., react quickly to changed circumstances, is the subject of Chapter 2, where we focus on the Less-than-truckload (LTL) industry segment. In the United States, this segment represents a $40 billion industry with about 25 major players. It handles shipments with a weight ranging between 120 and 10,000 pounds, more than what parcel carriers handle, but too small for full truckload transportation. To be profitable, LTL carriers have to consolidate shipments from different shippers so as to increase trailer utilization and minimize the "air" transported. In the last decade, the share of Less-than-Truckload within the trucking industry has grown as a result of economic trends, e.g., e-commerce with its aggressive service guarantees. The rise in B-to-B and B-to-C freight handled by LTL carriers has put additional pressure on daily operations. For instance, the use of weekly driver schedules, prepared in advance (typically a few days in advance for many major US carriers), provides little, if any, flexibility to adjust them on the day of operations. Given increasing day-to-day freight volume fluctuations, this is a major challenge, which is exacerbated by more and more aggressive service promises and a highly competition environment.

[5]https://www.bcg.com/publications/2020/understanding-why-agile-will-help-move-the-needle-post-covid-19

Thus, there is need for decision support technology that can, in near real-time, suggest changes to a load plan on the day of operations based on observed freight volumes. Fortunately, access to more and more real-time information regarding operations has become available, e.g., shipment volume and vehicle location information. Many, if not all, major players have invested heavily in equipping their facilities and vehicles with technology and intelligence that meets their needs for real-time information. Moreover, advances in computing power and infrastructure have made it more realistic to expect that near real-time optimization-based decision support is feasible. Another characteristic of LTL that justifies the need for a system wide optimization for the whole service network, is the fact that there is a central dispatch team that monitors and controls the operations for the entire network. The near real-time decision support methodology that we are proposing in this thesis will benefit the central dispatch as it has visibility over the entire network and optimizes for it, and thus, it will allow them to propagate the suggestions (shipment assignment to specific load, load cancellation, etc.) at the hub level multiple times during the day. In Chapter 2, we address this operational challenge by suggesting strategies to re-route shipments in the network in order to improve their on-time delivery performance without changing the planned driver schedules (i.e., planned loaded movements). These strategies are meant to be implemented in a decision support framework and used with a high cadence (typically every 30 minutes or hourly) on a daily basis to assign all the shipments that are in the network to the planned loads. The advantage of this approach is that it leverages the latest information about the status of all the shipments (potentially new unexpected ones) in the network and the scheduled direct movements to capture any recent changes and runs fast heuristics to find a good solution (i.e., assignment of shipments to planned loads) in a timely manner. Normally, in an ideal scenario when the observed demand volumes matches the forecast, these shipments will be shipped based on the *Planned Flow* information that dictates how shipments are routed in the network based on their origin and their final destination. Planned Flow is the product of *Service Network Design* problem that was studied extensively in literature (e.g., [5] and [6]). It gives an optimal routing of shipments in a consolidation network

based on an average day demand assumption. The rationale behind the planned flow is to facilitate the planning process in the hubs as shipments are grouped based on their final destination and systematically assigned to optimal direct movements. A secondary service network design problem is load planning and is used to determine the frequency of the daily loads that are needed to serve the demand for a given origin-destination pair. Load planning is what determines the *Load Plan*, which is the daily available capacity (i.e, number of direct movements to dispatch from each node in the network) required to satisfy the expected demand. The load plan generally covers a week and is updated on a daily basis. As demand is variable, the available capacity can be insufficient in some lanes and excessive in other lanes. For the case where the capacity is insufficient, the goal is to find for excess shipments alternative routes to the planned flow that can still deliver them on-time without having to invest in additional capacity. These alternative routes form the Alternate Flow information. The value of these alternative routes is highlighted in [7] as they can help absorb reasonable levels of uncertainty in the demand without having to increase the load capacity. We model this problem using a time-expanded network where the arcs are the existing timed loads. As the size of the problem is large, we suggest multiple heuristics to solve it fast and still reach good on-time performance.

The heuristics developed in Chapter 2, which adjust freight routes to best use available capacity when daily demand deviates from forecast demand, can likely be used to go a step further and suggest adjustments to available capacity, e.g., cancelling movements with very little freight, adding movements to avoid long, circuitous rerouted freight paths, and identifying "skip directs" - movements that bypass the cross-dock operation at a terminal. Such adjustments can save costs for logistics companies. For instance, suggesting to cancel a contractor schedule ahead of time can save its cost as the company will only incur a cancellation penalty without having to run the schedule. Even canceling a company's own driver schedule cycles can help release the resources and assign them to other tasks. Moreover, bypassing processing in intermediate hubs enables some shipments to reach their final destination earlier, saves resources in the hubs from unneces-

sary handling of shipments, and prevents trailers from occupying inbound and outbound doors and shipments from occupying space at the dock.

In the second part of this thesis, which spans Chapters 3 and 4, we focus on challenges encountered in the management of the ground equipment fleet in small package networks. The reusable equipment fleet of trailers and containers is essential to deliver the daily loads between each origin-destination pair of the service network. Our objective here is to ensure that the right equipment is available at the right time at the right location. As demand is naturally imbalanced between regions given the distribution of supply and demand, and given the variations in inbound and outbound load activity at a daily basis, some facilities in the network will see more inbound than outbound trailers possibly leading to a buildup of trailers that can exceed the facility yard capacity. Other facilities will see more outbound than inbound trailers possibly leading to equipment stock-outs and delays in executing planned freight movements. These two behaviors can alternate in the same facility where it sees equipment surplus that may exceed the number of yard spots at some times of the day, and equipment shortages at other times. This surplus and shortage can be detrimental on the daily operations. For example, an equipment shortage may lead to a load delay or cancellation that can be costly for the company as shipments may end up losing their tight service guarantee which is crucial in some of the companies' offerings to maintain customer satisfaction and competitive advantage. Another aspect that makes this planning problem hard is the heterogeneity in the type of trailers and containers used in the network. A heterogeneous fleet of equipment increases the complexity of equipment management as it destroys the self-balancing nature of driver circulations in the network, e.g., a driver can transport a 53-foot trailer from one location to another, but then return with two 28-foot trailers. Some companies prefer to use a more homogeneous fleet to avoid the complexity in operations. For instance, Amazon uses mainly the 53 foot trailers for their middle mile and doesn't use flatbeds and tankers[6]. Hours of service and union regulations may further complicate matters as it can result in (undesirable) bobtail movements, i.e., movements where a

[6]https://www.fool.com/investing/2020/06/24/in-midst-turmoil-amazon-flipped-delivery-switch.aspx

driver returns to his domicile in a tractor without pulling any trailer(s). This is due to the load and unload times that can vary a lot from one facility to another and depending on the time of day. In fact load and unload times depend a lot on the availability of the inbound and outbound doors, and the resources at the dock that can handle the shipments. As a driver schedule is constrained in time, a driver may not be able to wait for a long time for a trailer to be unloaded and made available and can return to his home location without pulling any trailer. To address equipment surplus or shortage at facilities, carriers essentially resort to repositioning equipment from facilities with a surplus to facilities with a shortage. This equipment repositioning can be costly especially if it is done at the day of operations as it requires to tender the empty trailer movements to third party carriers when the companies' own drivers are not available. Moreover, empty repositioning is one of the systematic mechanisms, if not the only, that network planners use to react to trailer unavailability in the network in day-to-day operations. To address regional imbalances, logistics companies often resort to using inter-modal logistics by sending empty trailers through one-way rail movements from one region to another (e.g., Northeast to Southwest). Companies can also resort to leasing equipment for short periods of time (typically during peak seasons such as Black Friday, Prime Day, Christmas holidays, etc.) or procuring additional fleet to align with a long term growing demand. All these solutions come at a significant cost. In Chapter 3, our goal is to ensure that each facility will have a minimum equipment imbalance during a planning horizon (typically a week in the future). We assume we have a large service network operating a heterogeneous fleet. We also assume a load plan is given with all the load movements that are planned to dispatch during the upcoming planning period of one week. For a given pair of facility and equipment type, we define the imbalance as the absolute value of the difference of total number of inbound loads and the total number of outbound loads that are using that equipment type. Our primary objective is to minimize the total imbalance for all facilities and equipment types by assigning the optimal equipment type to each load and finding the optimal repositioning plan. The rationale behind this weekly rebalancing is to ensure a facility is not systematically losing or gaining trailers that will

create imbalances in the network in the long run. This approach enables to maximize to balancing value of equipment type substitutions (i.e., changing an initially assigned equipment type of a load to a different equipment type). The substitutions between equipment types are not all possible and there are business rules that govern and constrain them. For instance, there are some facilities that are limited to only handle short equipment types and can't handle the traditional 53 foot trailers. Moreover, in some lanes, bobtail movements are not allowed for safety reasons. In Chapter 4, we shift the aim from minimizing the imbalance for a planning period to an inventory aware equipment management approach. While the imbalance minimization model strives to maintain the same inventory level for each equipment type at each facility and ignores the time related to inbound and outbound activity during the planning period, in contrast, the inventory aware model aims at monitoring the inventory at all facilities for all equipment types to avoid any occurrence of inventory stock-out or the violation of the yard capacity at any time of the planning period. This type of model allows for more flexibility as we can still enforce an ending inventory that can match the initial inventory, and thus satisfy the imbalance minimization objective of Chapter 3. Nevertheless, the problems become hard to solve as the size of the models increase due to the time dimension that is added to the problem. To keep the models tractable, a parsimonious time discretization for the time-expanded networks is required to control the size of the mathematical models. Furthermore, it is important to generate repositioning arcs as needed as opposed to generating the entire set of arcs that can be prohibitively large. To address this issue, we resort to a dynamic variable generation based algorithm that can find high impact repositioning arcs fast. The proposed inventory aware approach is dependent on the availability of near-real time information systems that track where the equipment fleet is located in the network (both equipment sitting in the yard and in-transit equipment). To have a good visibility on the equipment that is located at the facility premises, companies resort to intelligent Yard Management Service (YMS) tools that give the status of each trailer in the facility (unloading, loading, available, waiting to be relayed, inoperative, etc.). YMS tools generally rely on GPS and RFID technology to automate the tracking

process and optimize the operations by providing real-time information about the available assets[7]. Such technology can be used to also track in-transit equipment and forecast when it will arrive to destination.

In summary, in this thesis we propose algorithmic ideas for solving (some) large-scale problems faced by logistics companies. There are multiple challenges when solving such problems. First, there is often a constraint on the run-time. As some of those problems concern day-to-day operations and need to be solved frequently during the day to account for a continuous stream of information about the network state, we need to find a solution and implement it in a short amount of time (e.g., 30 minutes to one hour). Second, the size of the instances is often large as the service networks we seek to provide solutions for are large. Moreover, our objective is to find a system-wide solution as the decision making is generally centralized for these networks. Solving such large instances with exact methods is virtually impossible. This justifies why we resorted to fast heuristics that use insights from the structure of the problems. Last, these problems can be shown to belong, in their most general form, to the class of NP-hard problems. Nevertheless, we can also show that there are special settings in which some of these problems are polynomially solvable. This knowledge can suggest fast solution approaches.

Remark

This thesis is one of the outcomes of a collaborative research project of the Transportation Analytics and Operations Research team at *UPS Supply Chain Solutions* and a team of faculty and PhD students at the School of Industrial and Systems Engineering at Georgia Institute of Technology. The research described Chapter 2 also involved Ian Herszterg and the research described in Chapter 3 also involved Yu Yang. However, the parts presented in this thesis are those for which I had primary responsibility.

[7] https://www.questsolution.com/blog/item/43-developing-a-more-efficient-yard-management-strategy

CHAPTER 2
NEAR REAL-TIME SHIPMENT LOADING FOR LESS-THAN-TRUCKLOAD CARRIERS

2.1 Introduction

In the LTL segment, service providers are often faced with the operational challenge regarding how actual shipments entering the network at the day of operations need to be efficiently routed to their destinations with the existing load capacity that was already planned ahead of time. The line-haul network is assumed to be established and designed to cost-effectively deliver forecasted volumes while satisfying the service guarantees. Driver schedules are also assumed to be created a week ahead. Moreover, average origin/destination daily volumes are used to determine how many trailers are needed to satisfy the demand between two facilities of the service network. As picked-up shipments and customer reservations are known gradually during the day, these quantities can be different from the volumes expected during that day. A natural question that arises is: can we still deliver the realized demand with the existing trailer loads while satisfying the promised service guarantees? If not, to which extent can we minimize the lateness of shipments by adjusting their routes without modifying the load plan (i.e., adding new loads or canceling planned ones)? In this chapter, we discuss the design and implementation of several efficient heuristics for re-routing shipments to their destination without altering driver schedules (i.e., planned vehicle movements) so as to improve on-time performance.

Currently, shipments arriving at a terminal are typically handled in a *first in, first out* (FIFO) fashion. Terminal operators assign arriving shipments to the trailer on the planned path to the shipment's destination with the earliest (feasible) departure time in order to avoid accumulation of (too many) shipments in front of dock doors and to get the shipment closer to its destination as soon

as possible. Such a strategy helps achieve performance targets set for the terminal, but ignores how these decisions affect the performance of the entire system. Furthermore, such a strategy facilitates accommodating last minute customer reservations and unexpected shipment arrivals (e.g., sent by other terminals that had to deal with an unexpectedly high freight volume). Our heuristics seek to find the right balance between local and system-wide performance.

The contributions of the research discussed in this chapter can be summarized as follows:

- We introduce a variety of metrics related to service quality that help quantify the benefits of real-time adjustments to loadplans;
- We present a number of fast heuristics to suggests loadplan adjustments that substantially improve on-time performance;
- We demonstrate the practical viability and value of these heuristics in a computational study using real-life data from a large US LTL carrier.

The remainder of the chapter is organized as follows. In Section 2.2, we formally define the problem, present a formulation of the problem, and discuss the scope and goals of the research. In Section 2.3, we introduce different heuristics to solve realistic, large-size instance of the problem. In Section 4.6, we summarize and analyze the results of a set of computational experiments. In Section 4.7, we highlight the practical value of the proposed heuristics and discuss ongoing research to further enhance the technology.

2.2 Problem Statement

LTL carriers seek to achieve high trailer utilization, especially when moving trailers over longer distances, by employing a hub-and-spoke network, represented schematically in Figure 2.1. After being collected from customers, shipments are brought to an *End-of-Line* terminal, where they are sorted, consolidated, and dispatched to a *Breakbulk* terminal. Breakbulk terminals serve as central

hubs for a region and consolidate shipments from different End-of-Line terminals in order to ensure high load factors (i.e., high utilization) for trailers departing to other Breakbulk terminals. Shipments in trailers arriving at a Breakbulk terminal are sorted, consolidated, and either dispatched to another Breakbulk terminal or to an End-of-Line terminal in the region for final delivery. For a more detailed description of LTL operations see [8].

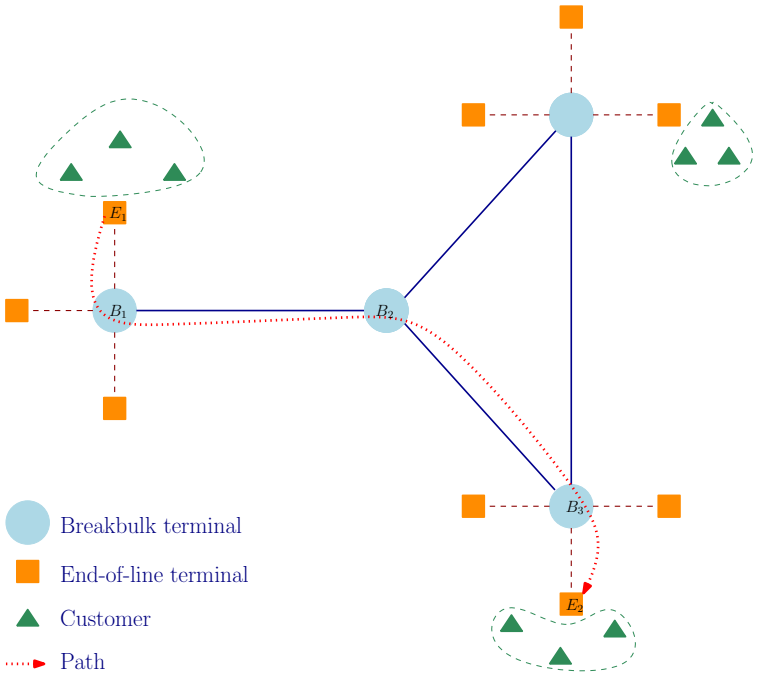


Figure 2.1: A hub-and-spoke LTL network. The dotted red path represents a load from End-of-Line E1 to End-of-Line E2 involving four schedule legs $E_1 \rightarrow B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow E_2$.

Consolidation carriers have traditionally focused their decision support efforts on planning rather than execution. In load planning, given a forecast of freight volumes between origins and destinations, the goal is to identify origin-destination paths that meet service guarantees, that are likely to lead to effective consolidations, and that are not too costly, i.e., that provide the right balance between route circuity and trailer utilization.

An origin-destination path is defined by a sequence of (intermediate) terminals (possibly empty) where shipments are unloaded from one trailer and loaded into another trailer. The set of all origin-destination paths is commonly referred to as the *planned flow*. The planned flow dictates where a

shipment is send next, given that it becomes available at a specific terminal at a specific time of day. If actual freight flows match forecast freight flows, then using the planned flow paths should minimize cost while ensuring service guarantees are met. However, in practice, actual freight flows rarely match forecast freight flows, and it may not be possible to meet service guarantees relying solely on the planned flow paths. Therefore, in practice, a set of alternate original-destination paths is constructed and used when capacity on a planned flow path is insufficient. The trailer movements are not affected and are executed as planned, but the path from origin to destination for (some) shipments is changed. The set of all alternate origin-destination paths is commonly referred to as the *alternate flow*.

There is abundant literature on network design for the LTL trucking industry. However, there is scant literature on near real-time dynamic load planning for service networks. A taxonomy of service network planning has been introduced in [9] and is widely used to delineate strategic, tactical, and operational planning. The bulk of the literature focuses on strategic and tactical planning, and only a few papers consider operational planning, which is the topic of our research. To the best of our knowledge, no prior work has investigated the use of alternative paths to handle daily demand fluctuations in an operational setting. The value of introducing (planned) alternative paths in a service network to hedge against demand uncertainty has been highlighted in [7]. The authors demonstrate that it is sufficient to have a single alternative option at the terminals visited along an origin-destination path to absorb most of the demand uncertainty. Due to the fact that integer programming formulations (based on a network or a time-expanded network representation) of tactical service network design problems are difficult to solve, especially for realistic instance sizes, most solution approaches are heuristic. The survey papers by [5] and [6] summarize much of the research in this area. Examples of papers proposing meta-heuristics include [10], [11], [12], [13] and [14], proposing Lagrangean heuristics include [15] and [16], and proposing slope scaling heuristics include [17].

The importance of using near real-time information to optimize freight transportation opera-

tions on the day of operations is highlighted in [18], [19], [20], [21], [22], [23], [24], [25], and [26]. [27] discuss the benefits and difficulties associated with implementing near real-time optimization models in the motor carrier industry, stressing the importance of acknowledging the human factor in decision making. The survey paper on intelligent freight network systems by [28] includes an overview of the opportunities, but also the challenges, that access to real-time information offers. The authors argue that operations research techniques need to be leveraged to bring additional value to motor carriers and increase their agility in the modern fast-paced environment. In the aforementioned papers, different types of formulations are used to model network design problems. Arc-based formulations (in which an arc represents a direct trailer movement between two terminals) are suggested by [13], [15], and [16]. Path-based formulations (where a path represents a sequence of direct movements that take a shipment from its origin to its destination) are suggested in [29] and [12]. Tree-based formulations (where a tree represents direct movements that take shipments from their origin to a particular destination) are suggested in [30] and [17].

In a setting more similar to the one we consider, [31] propose an integer programming based look-ahead formulation to solve a dynamic load planning problem. The formulation is based on the current state of the network, i.e., the shipments in the system, the shipments forecast to enter the system, the set of drivers and their assigned trailer movements, and is solved using a type of "relax-and-fix" approach, in which the formulation is decomposed into multiple subproblems based on a discretization of the time horizon, and subproblems are solved in order of time. The solution approach is tested on data from a reasonably large LTL carrier in the U.S. (with a network of about 300 terminals with about 40,000 daily shipments) and shows positive results in terms of trailer utilization and on-time delivery performance.

We focus on the design and implementation of decision technology to route shipments on their planned or alternate paths given the latest information on the freight already in or anticipated to enter the service network so as to ensure the highest possible on-time performance. The technology will support central dispatchers, in near real-time, when they are faced with deviations

from expected freight volumes. When the decision technology is invoked it considers three sets of shipments: shipments that have been picked up and are available for dispatch at a terminal at the start the planning period, shipments that are en-route at the start of the planning period and are expected to reach their next intermediate terminal within the planning period, and shipments that are expected to be available for dispatch at a terminal at some later time during the planning period (referred to as forecast shipments). Given the trailer movements that are to be executed during the planning period, we may find that all shipments can be delivered at their destination on-time using the planned flow, or some shipments will be delivered late if only the planned flow is used.

We assume a planning period of 48 hours starting at a specific time during the day, typically 6pm. The choice of the length of the planning period is motivated by the fact that the fastest growing offerings of freight transportation companies are next-day and two-day delivery. Consequently, the majority of freight in the system will reach its final destination within 48 hours. In the remainder, for convenience and ease of presentation, we assume that there are no late shipments at the start of the planning period, and that for each shipment there is a (unique) planned path along which the shipment can reach its final destination on time (if it would be the only shipment in the system and it would not have to compete for capacity). We also assume that planned flow and alternate flow paths are given, and that scheduled capacity is fixed (i.e., there is a given number of planned trailers with known dispatch and arrival times between pairs of terminals). The scheduled capacity is such that if the anticipated daily demand realizes the capacity is sufficient to move all the shipments along their planned paths.

To model the problem, we use a time-expanded directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. Let $\mathcal{S}$ be the set of shipments and let $\mathcal{L}$ be the set of trailer movements during the planning period. Each shipment $s \in \mathcal{S}$ has an associated quantity qty_s, origin terminal org_s (where it enters the network), origin time otm_s (when it enters the network), destination terminal dst_s (where it needs to end up), and due time due_s (when it needs to end up there). Each trailer movement $l \in \mathcal{L}$ has an associated capacity cap_l, origin terminal org_l, departure time at the origin dtm_l, destination terminal dst_l, and

arrival time at the destination atm_l. All times are relative to the start of the planning period. A node $(u, t) \in \mathcal{V}$ represents a location u at a point in time t. The node set $\mathcal{V}$ is partitioned into three subsets: $\mathcal{V}_D$, $\mathcal{V}_A$, and $\mathcal{V}_E$. The node sets $\mathcal{V}_D$ and $\mathcal{V}_A$, respectively, represent the departures and the arrivals of trailers, and the node set $\mathcal{V}_E$, with nodes $(u, +\infty)$ – one for each terminal u, represents the end of the planning period. The arc set $\mathcal{A}$ is also partitioned into three subsets: $\mathcal{A}_L, \mathcal{A}_H$, and $\mathcal{A}_E$. The arc set $\mathcal{A}_L$ represents the trailer movements in $\mathcal{L}$. An arc $((u, t), (u, t + 1)) \in \mathcal{A}_H$ models the possibility for shipments to remain at terminal u from a trailer arrival or departure time t to the next trailer departure time $t + 1$, where the last "holding" arc is $((u, t), (u, \infty))$. An arc $((u', \infty), (u, \infty)) \in \mathcal{A}_E$ models a shipment being at an intermediate terminal u' at the end of the planning horizon and its transfer from u' to its destination terminal u. An arc $((u', t), (u, \infty)) \in \mathcal{A}_E$, for any terminal u' and t after the end of the planning horizon, models the situation where a shipment is en-route at the end of the planning horizon, reaches terminal u' at time t, and is transferred from u' to its destination terminal u. Figure 2.2 shows a part of a time-expanded network for shipments destined to terminal C.

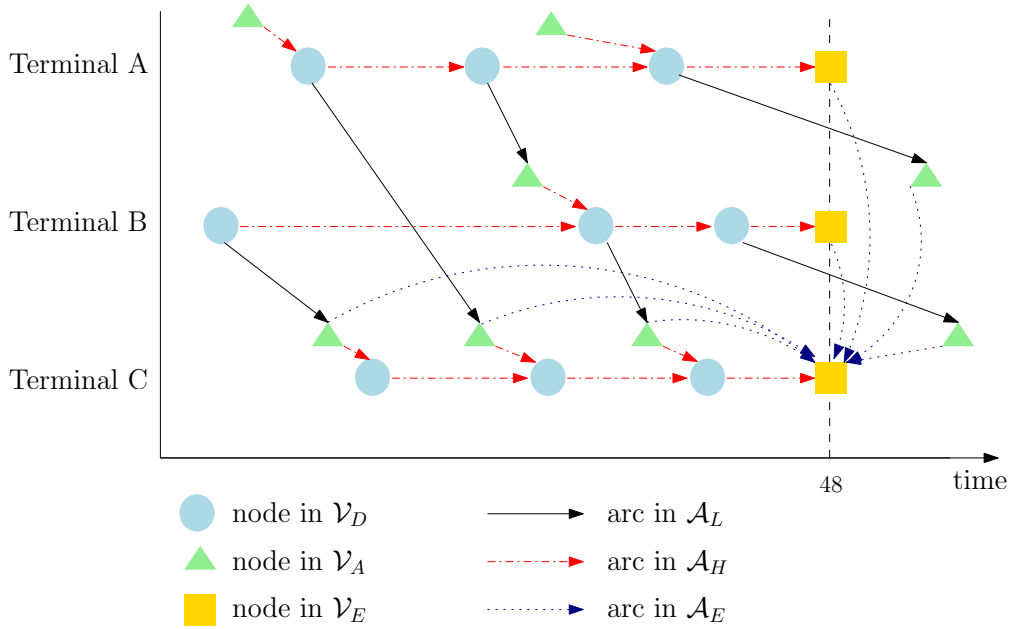


Figure 2.2: Part of a time-expanded network for shipments destined to Terminal C.

Each shipment enters the system at a source node $(org_s, otm_s) \in \mathcal{V}_D$ and leaves the system at a sink node $(dst_s, +\infty) \in \mathcal{V}_E$. Let $a = (u, v)$ with $u, v \in \mathcal{V}$ denote an arc in $\mathcal{A}$. Let $\mathcal{A}(s)$ denote the set of feasible arcs for shipment s, i.e., the relevant arcs in $\mathcal{A}_L$ along the planned and the alternate paths for s and the relevant arcs in $\mathcal{A}_H$ and $\mathcal{A}_E$ at terminals along the planned and the alternate paths for s.

Let x_a^s denote a binary decision variable representing assigning of shipment $s \in \mathcal{S}$ to arc $a \in \mathcal{A}(s)$ ($x_a^s = 1$) or not ($x_a^s = 0$). Then the dynamic load planning problem can be formulated as an arc-based multi-commodity flow problem on $\mathcal{G}$ as follows:

$$
\begin{aligned}
(\mathcal{ABF}) \min \quad & \sum_{s \in \mathcal{S}} \left[\Big(\sum_{a \in \mathcal{A}(s)} c_a^s x_a^s \Big) - due_s \right]_+ \\
\text{s.t.} \quad & \sum_{\substack{s \in \mathcal{S} \\ a \in \mathcal{A}(s)}} qty_s x_a^s \leq cap_a \; \forall a \in \mathcal{A}_L && (2.1) \\
& \sum_{\substack{w \in \mathcal{V} \\ (u,w) \in \mathcal{A}(s)}} x_{(u,w)}^s = \sum_{\substack{w \in \mathcal{V} \\ (u,w) \in \mathcal{A}(s)}} x_{(w,u)}^s \; \forall s \in \mathcal{S}, u \neq (org_s, otm_s), u \neq (dst_s, +\infty) && (2.2) \\
& \sum_{\substack{w \in \mathcal{V} \\ ((org_s, otm_s), w) \in \mathcal{A}(s)}} x_{((org_s, otm_s), w)}^s = 1 \; \forall s \in \mathcal{S} && (2.3) \\
& \sum_{\substack{w \in \mathcal{V} \\ (w, (dst_s, +\infty)) \in \mathcal{A}(s)}} x_{(w, (dst_s, +\infty))}^s = 1 \; \forall s \in \mathcal{S} && (2.4) \\
& x_a^s \in \{0, 1\} \; \forall s \in \mathcal{S}, a \in \mathcal{A}(s),
\end{aligned}
$$

where cap_a represents the capacity of arc $a \in \mathcal{A}$, i.e., the capacity of the trailer movement, and c_a^s represents an appropriately chosen time-related cost for sending shipment s along arc $a \in \mathcal{A}(s)$. More specifically, let $EstArrival(s, o, t)$ be the estimated arrival time of shipment s at its destination given that it will be available at terminal o at time t, will follow its planned path, and will use the earliest possible trailer movements along the planned path. Then the time-related cost

c_a^s is given by:

$$c_a^s = \begin{cases} 0, & \forall\, a = ((u, t_u), (v, t_v)) \in (\mathcal{A}_L \cup \mathcal{A}_H) \cap \mathcal{A}(s), v \neq dst_s \\ t_v, & \forall\, a = ((u, t_u), (v, t_v)) \in \mathcal{A}_L \cap \mathcal{A}(s), v = dst_s \\ EstArrival(s, u, 48), & \forall\, a = ((u, +\infty), (dst_s, +\infty)) \in \mathcal{A}_E \\ EstArrival(s, u, t), & \forall\, a = ((u, t), (dst_s, +\infty)) \in \mathcal{A}_E \text{ with } t > 48 \end{cases}$$

The objective in $\mathcal{ABF}$ seeks to minimize the total lateness of shipments. No incentive is given for delivering shipments early, i.e., it suffices to deliver a shipment at or before its due time. If a shipment that does not reach its destination during the planning period, its path ends with an arc $((u, \infty), (dst_s, \infty)) \in \mathcal{A}_E$. Consequently, the model seeks to get such shipments as close as possible to their destination (as it captures the projected arrival time at the destination after the end of the planning period). Constraints (1) ensure that trailer movements do not exceed their capacity and Constraints (2), (3), and (4) ensure flow conservation for shipments (i.e., that there is a unique origin-destination path for each shipment).

The dynamic load planning problem can also be formulated as a path-based multi-commodity flow problem. Let $\mathcal{P}(s)$ denote the set of feasible paths for a given shipment $s \in \mathcal{S}$. A feasible path for s is a path in the time-expanded network $\mathcal{G}$ representing a time-feasible sequence of trailer movements from the shipment's origin to its destination. The objective, as before, is to minimize total lateness. Let c_p^s denote the lateness of shipment $s \in \mathcal{S}$ when using feasible path $p \in \mathcal{P}(s)$. Let x_p^s denote a binary decision variable representing assigning shipment $s \in \mathcal{S}$ to path $p \in \mathcal{P}(s)$.

The path based formulation is as follows:

$$(\mathcal{PBF}) \quad \min \sum_{s \in \mathcal{S}} \sum_{p \in \mathcal{P}(s)} c_p^s x_p^s$$

$$s.t. \quad \sum_{p \in \mathcal{P}(s)} x_p^s = 1 \; \forall s \in \mathcal{S} \tag{2.5}$$

$$\sum_{p \in \mathcal{P}} \sum_{\substack{s \in \mathcal{S} \\ p \in \mathcal{P}(s)}} \delta_{ap} qty_s x_p^s \leq cap_a \; \forall a \in \mathcal{A} \tag{2.6}$$

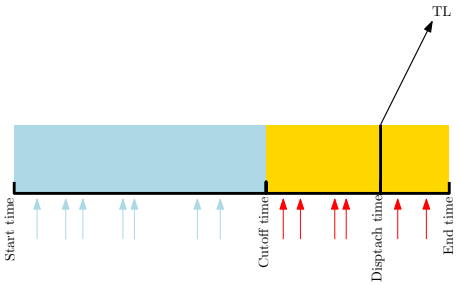
$$x_p^s \in \{0,1\} \; \forall s \in \mathcal{S} \, \forall p \in \mathcal{P}(s),$$

where δ_{ap} is an indicator set to 1 when arc a belongs to path p and 0, otherwise. Constraints (1) ensure that each shipment is assigned a (unique) feasible path. Constraints (2) ensure that trailer movements do not exceed their capacity.

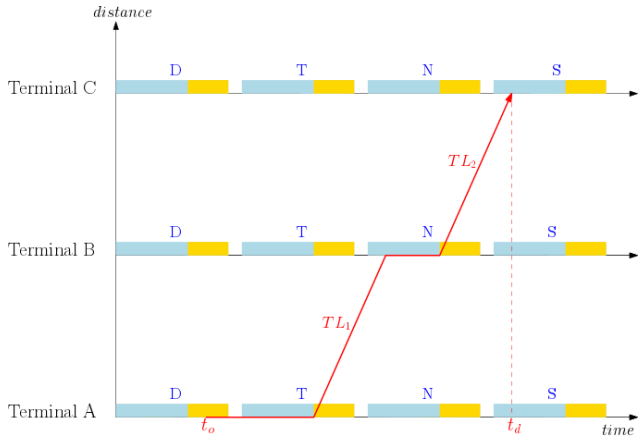
Solving either the arc-based or the path-based formulations for large-scale instances in a short amount of time is virtually impossible. For instance, if we assume a network of 300 terminals, 10,000 trailer movements, and 75 thousand daily shipments, the arc-based formulation will yield a model with 1.5 billion binary variables and 45 million constraints. Regardless of how the set of feasible paths for a shipment is constructed, the path-based formulation too, will have a prohibitively large number of variables and constraints. As the problem is meant to be solved several times during the day of operations, finding a solution should take no more than 10 to 15 minutes. Thus, the use of efficient and effective heuristics is the only viable option.

Another layer of complexity stems from the use of *sorts* to simplify and manage terminal operations. A sort is a time period at a terminal during which shipments are handled and loaded in trailers that will transport them to their next destination. The concept of a sort originated in the small package business where parcels go through a conveyor sortation station in order to be consolidated with other parcels and directed to their respective dock doors. It has been adopted by some carriers in their LTL freight business as well. The idea behind a sort is that all shipments that

arrive at the terminal before the cut-off time associated with the sort will be processed during that sort and will be ready to be dispatched by the end of that sort. Shipments that arrive after the cut-off time associated with a sort are held and will be processed in the next sort. Sorts will be considered in the methodologies presented in the next section and provide a mechanism for capturing and handling different organization of operations at terminals. In our motivating setting, there are four sorts within a day of operations: *Day*, *Twilight*, *Night*, and *Sunrise*. Sort-based strategies seek to minimize the number of shipments held (or rolled over) to subsequent sorts. An example of the organization and use of sorts is given in Figure C.2. Cut-off and processing times at terminals can be handled by adjustments to the arcs representing trailer movements, i.e., an arrival after a cut-off time is mapped to the start time of the next sort and processing times are incorporated in the trailer movement times. Thus, for the remainder, we assume that all trailers arriving in a sort arrive before the cut-off time and shipments are available for dispatch immediately upon arrival.



(a) Type of the activities in a sort. Blue arrows represent shipments that arrive before the cutoff time and can be handled in the sort. Red arrows represent shipments that arrive after the cutoff time and will be rolled over to the next sort



(b) Example of a shipment delivered through a sequence of terminal sorts from its origin A to its final destination C. t_o represent the time the shipment was made available at terminal A, t_d represents the arrival time at terminal C

Figure 2.3: Example of the organization and use of sorts.

2.3 Methodology

To determine whether adjustments to the loadplan are advantageous given up to date information on the shipments in and entering the service network, we seek to find time-feasible paths for all

shipments such that the total (expected) lateness is minimized. That is, given the planned trailer movements, we find a path for each shipment during the planning period, using only planned or alternate flow options, that minimizes the total expected lateness, where, for shipments reaching their destination during the planning period, we will know the actual lateness, and for shipments that do not reach their destination during the planning period, we use an optimistic estimate of their lateness. Given the size of problem, i.e., the large number of planned trailer movements in the planning period and the large number of shipments in or entering the service network during the planning period, and the limited time available, i.e., at most 15 minutes of computing time, we develop greedy, but intelligent, trailer loading heuristics that balance the need for efficiency with the desire for quality.

We start by presenting a baseline heuristic that is shipment focused and assumes that shipments in a sort at a terminal are processed first-in, first-out (FIFO). Although naive, it reflects the viewpoint, which has been popular in practice, that it is beneficial to keep freight moving in the direction of its final destination. Next, we present a heuristic that is trailer movement focused and processes trailer movements in non-decreasing order of departure times. After observing that a simple upfront analysis of the trailer movements can identify blocks of trailer movements that can be considered together, we present a heuristic that is block focused and processes blocks in non-decreasing order of departure times of the first trailer movement in a block.

2.3.1 FIFO Loading

For a given shipment $s \in \mathcal{S}$ let P_s denote the {terminal, sort} pair where s has become available for dispatch. This can be at the terminal where the shipment enters the linehaul system or at an intermediate terminal on the shipment's journey from origin to destination. We first check to see if there is a trailer movement departing in P_s after the arrival of s on the planned path for s that has capacity remaining to accommodate s. If such a trailer movement exists, we load s in the first such trailer movement, i.e., the one with the earliest departure time, and update P_s (i.e., we set P_s

to the {terminal, sort} pair defined by the destination of the trailer movement and the sort in which it arrives). If no such trailer exists, we check to see if there is a trailer movement departing in P_s after the arrival of s on an alternative path for s that has enough remaining capacity to accommodate s. If such a trailer movement exists, we load s in the first such trailer movement and update P_s. Finally, if no such trailer movement exists either, we hold s until the next sort (we update P_s accordingly). Heuristic FIFO-PUSH processes shipments in $\mathcal{S}$ non-increasing order of arrival time (at their current location). When a shipment arrives at its destination, it is not reinserted in $\mathcal{S}$, which ensures that the heuristic terminates after a finite number of steps. Algorithm 1 shows the pseudo-code for FIFO-PUSH.

Algorithm 1: FIFO-PUSH

$\mathcal{S} \leftarrow$ list of all shipments in the network, sorted by arrival time

$\mathcal{L} \leftarrow$ list of all trailers in the network departing during the time horizon

for *each* *shipment* *s* *in* $\mathcal{S}$ do

- $P_s \leftarrow$ pair {terminal, sort} where s is available for pickup
- $L_s^P \leftarrow$ subset of trailers in $\mathcal{L}$ departing during P_s going though the planned path for s
- $l_P \leftarrow$ earliest feasible trailer in L_s^P , with enough capacity left to load s
- **if** $l_P \neq \emptyset$ **then**
  - load s in l_P and update $\mathcal{S}$
- **else**
  - $L_s^A \leftarrow$ subset of trailers in $\mathcal{L}$ departing during P_s going through one of the alternative paths for s
  - $l_A \leftarrow$ earliest feasible trailer in L_s^A, with enough capacity left to load s
  - **if** $l_A \neq \emptyset$ **then**
    - load s in l_A and update $\mathcal{S}$
  - **else**
    - hold s and postpone the loading decision to the next sort at the terminal

2.3.2 Urgency Loading

The second heuristic focuses on trailer movements rather than shipments and processes trailer movements in order of nondecreasing departure times. For a given trailer movement, we have to decide which of the available shipments to load in that trailer movement. As our goal is to minimize total lateness, we use the *urgency* of a shipment as the basis for making loading decisions.

Consider, again, the function $EstArrival(s, o, t)$, which estimates the arrival time at destination terminal dst_s of a shipment s that is currently at terminal o, that departs from that terminal

after time t, that follows its planned path, and that uses the earliest possible trailer movements along the planned path. We define $Urgency(s, o, t)$ as the urgency of shipment s given that it is currently at terminal o at time t:

$$Urgency(s, o, t) = EstArrival(s, o, t) - due_s. \tag{2.7}$$

A positive value of $Urgency(s, o, t)$ means that the shipment will be late even in the best case scenario, i.e., that it can follow its planned path and can always depart on the earliest trailer movements along the path. A non-positive value means that the shipment is expected to arrive on time at its destination.

For convenience, we will, in the remainder, refer to a trailer movement simply as a trailer. Let $\mathcal{L}$ be the set of trailers departing within the planning period in order of non-decreasing dispatch times. For a given trailer $l \in \mathcal{L}$, let $\mathcal{S}_l$ be the set of shipments available for loading at org_l, i.e., every shipment in $\mathcal{S}_l$ has arrived at org_l before dtm_l. Let $\mathcal{S}_l^P$ be the subset of *planned path shipments* in $\mathcal{S}_l$, i.e., the subset of shipments for which l is on the planned path. Let $\mathcal{S}_l^A$ be the subset of *alternate path shipments* in $\mathcal{S}_l$, i.e., the subset of shipments for which l is on the alternate path. We start by loading l with the shipments in $\mathcal{S}_l^P$ in order of non-increasing urgency, and, in case of ties, in order of non-increasing size until there is no capacity left in l and/or all shipments in $\mathcal{S}_l^P$ have been loaded. If l has any remaining capacity after processing shipments in $\mathcal{S}_l^P$, we repeat the loading process with the shipments in $\mathcal{S}_l^A$, again loading in order of non-increasing urgency, and, in case of ties, in non-increasing order of size, before moving on to the next trailer. Shipments not loaded during a sort will naturally be processed in subsequent sorts. Algorithm 2 gives the pseudo-code for URG-PULL. URG-PULL focuses on moving shipments towards their destination as early as possible and on using as much of the available capacity as possible. Both are "rules of thumb" often used in practice.

Algorithm 2: URG-PULL

$\mathcal{L} \leftarrow$ list of all trailers in the network departing during the time horizon, sorted by dispatch time

for ***each*** *trailer* l *in* $\mathcal{L}$ **do**

  $\mathcal{S}_l^P \leftarrow$ list of planned path shipments available for pickup at the origin of l, sorted by urgency then by quantity, both in descending order

  while *there is capacity left in* l *and* $\mathcal{S}_l^P \neq \emptyset$ **do**

    load shipments from $\mathcal{S}_l^P$ in l

  if l *has capacity left* **then**

    $\mathcal{S}_l^A \leftarrow$ list of alternative path shipments available for pickup at the origin of l, sorted by urgency then by quantity in descending order

    while *there is capacity left in* l *and* $\mathcal{S}_l^A \neq \emptyset$ **do**

      load shipments from $\mathcal{S}_l^A$ in l

URG-PULL can be implemented efficiently. The trailers are processed one by one in order of nonincreasing departure times. The shipments are maintained in unordered lists, one for each terminal. Initially, a shipment is placed in the list associated with the terminal where it enters the system. Whenever a shipment is loaded and dispatched in a trailer, we update the lists associated with the origin and destination of the trailer by removing the shipment from the list of shipments at the trailer's origin and inserting it in the list of shipments at the trailer's destination. Processing a trailer involves going through the list of shipments at its origin, which takes linear time. If a shipment is loaded onto a trailer, deleting it from the list at the origin and inserting it in the list at the destination takes constant time.

Even though moving shipments towards their destination as early as possible is desirable, using alternate flow to do so may not necessarily be best. Therefore, we also consider the variant URG-PULL-PF which only loads shipments on their planned flow path. Similar to URG-PULL, we sort trailers in order of nondecreasing departure times and process them one after the other. Each

trailer l is loaded with planned path shipments only (i.e., with shipments in $\mathcal{S}_l^P$). Even when l has a remaining capacity, no additional shipments are loaded (i.e., shipments in $\mathcal{S}_l^A$ are ignored).

2.3.3 Block Loading

The myopic nature of URG-PULL, which attempts to move shipments closer to their destination whenever possible, may result in the use of unnecessarily many alternate paths, which, in turn, may results in too many shipments arriving at a terminal at a time when that terminal does not have enough outbound trailer capacity available to send these shipments on towards their final destination. This reflects the fact that URG-PULL considers only one trailer at a time and does not look ahead. Next, we introduce *block loading*, which considers a number of consecutive trailers departing from a terminal simultaneously, which, therefore, addresses one of the limitations of URG-PULL. Furthermore, we consider several variants of block loading in which we incorporate different look-ahead strategies, which, therefore, addresses another limitation of URG-PULL.

Let $\mathcal{PTS}$ be the list of {terminal, sort} pairs in the planning period. Given a pair $\{t,\ s\}$ in $\mathcal{PTS}$, let K denote the set of shipments available at terminal t at the start of the sort s plus any forecast shipments that will become available for loading during sort s (i.e., before the cut-off time of the sort). Let L^D denote the set of outbound trailers departing from t during sort s and L^A be the set of inbound trailers arriving at t during sort s. If $L^A = \emptyset$, then *every* shipment available for loading during the sort is known and one could solve a single optimization problem that assigns the shipments in K to the trailers in L^D. However, when $L^A \neq \emptyset$, trailers will arrive during the sort and some of the shipments available for loading during the sort are not yet known, namely those that arrive on the trailers in L^A. Consequently, solving a single optimization problem that assigns the shipments in K to the trailers in L^D is no longer advisable, as only partial information about the shipments available for loading during the sort is available. The two insights that underpin block loading are (1) that the information regarding shipments available for loading during a sort depends on the order in which we process the {terminal, sort} pairs in $\mathcal{PTS}$, and (2) that by partitioning

sorts into blocks (of trailers), and by processing these blocks in a specific order, we can ensure that *every* shipment available for loading during a block is known.

More specifically, we define a block as the set of consecutive outbound trailer departures between two consecutive inbound trailer arrivals within a sort (where the start time of the sort and the end time of the sort are also considered inbound trailer arrivals). In other words, a block is the largest possible set of departing trailers within a sort such that all shipments that can be loaded in these trailers are known at the time of the first trailer departure. Therefore, we can solve a single optimization problem to assign these shipments to the trailers in the block. We distinguish two approaches for creating blocks: (a) *static* generation of blocks, which generates blocks as described above (i.e., using the arrival times of inbound trailers), and (b) *dynamic* generation of blocks, where knowledge of the order in which blocks are processed is used to expand the size of some of the blocks by recognizing that for some of the blocks the shipments in the inbound trailer defining the end of the block are known by the time the block will be processed. In the following subsections, we describe in more detail how blocks are created and how shipments are assigned to trailers in a block.

Static generation

We start by observing that for a set of consecutive trailer departures at a terminal occurring between two consecutive trailer arrivals at that terminal, all the information about the shipments that can be loaded onto these trailers is known at the time of the departure of the first trailer (in fact at the time of the arrival of the trailer that precedes it). We call such a set of trailers a *block*. Importantly, we can assign the shipments available at the start of the block to the trailers in the block simultaneously, rather than one by one, which may be beneficial. We consider the start and end of a sort as "trailer arrivals" to ensure that every block occurs within a sort. It is easy to generate these blocks in nondecreasing order of the departure time of the first trailer in the block.

In the static generation of blocks, we iterate over every pair in $\mathcal{PTS}$ and create groups of departing trailers delimited either by the arrival time of an inbound trailer after the start of the sort or by the end time of the sort. More precisely, blocks are limited in size by the length of the sort at the terminal, i.e., no block contains trailers departing in more than one sort. The start and end times of the planning horizon are also used as time points for delimiting a block. Figure 2.4 shows

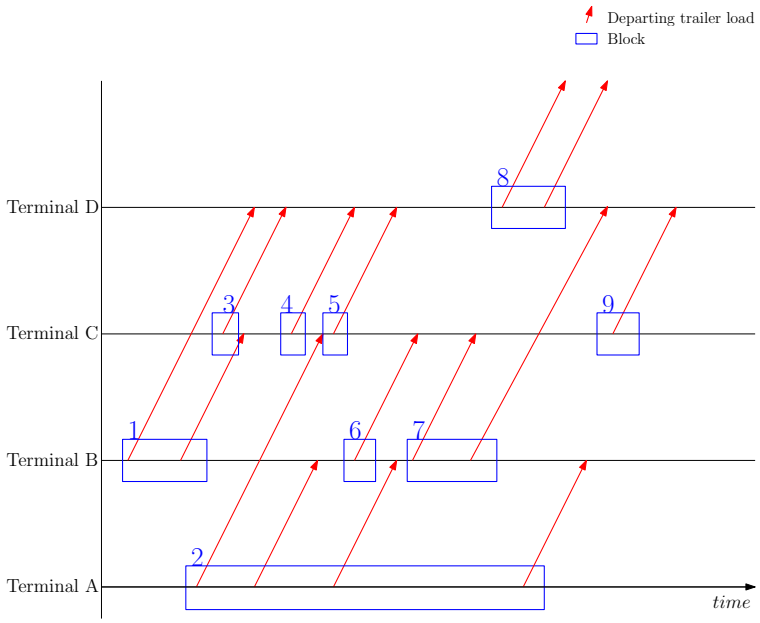


Figure 2.4: An example of the static generation of blocks. The numbers at the top left corners of the blocks represents the order in which the blocks are created.

an example of the static blocks created at different terminals. In this example, we assume a single sort covering the entire planning period at all terminals. Terminal A has a single block, as there are no inbound trailers, during the planning period. Algorithm 3 gives the pseudo-code for static block generation.

Dynamic generation

In static block generation, blocks can end up being smaller than they can be. This is the case when the shipments arriving in a trailer that would define the end of a block are already known by the time we start creating the block. This happens when the block that contains the (departure of such)

Algorithm 3: Static Block Generation

$Terminals \leftarrow$ list of all the terminals in the network
$\mathcal{B} \leftarrow \{\}$
for ***each*** *terminal* org *in* $Terminals$ **do**
  $lastBlock \leftarrow false$
  $t_b \leftarrow 0$
  while $lastBlock$ *is* $false$ **do**
    $t_0 \leftarrow$ dispatch time of first trailer that departs at time $t \geq t_b$ at terminal org
    $t_1 \leftarrow$ arrival time of first trailer that arrives at time $t > t_0$
    $t_b \leftarrow \min\{$end of current sort, $t_1\}$ at terminal org
    $b \leftarrow$ list of all trailers that dispatch in the time window $[t_0, t_b)$ at terminal org
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{b\}$
    if *there are no departing trailers after* $t \geq t_b$ **then**
      $lastBlock \leftarrow true$
Return $\mathcal{B}$

trailer has already been processed. For example, in Figure 2.4, we see that the trailer that defines the end of Block 3 has already been processed during the creation of Block 1 and thus its shipments are known. With this observation, it may be possible to create larger blocks as follows. In dynamic block generation, when creating blocks, we will use the most up-to-date information to determine the end of each block. Let $\mathcal{L} = \{l_1, \ldots, l_m\}$ be the set of trailers dispatched during the planning period in nondecreasing order of departure times. After we initiate a new block and the terminal associated with the block is known, we continue to add trailers from $\mathcal{L}$ that depart at that terminal to the block until we encounter a trailer that arrives at that terminal but has no shipments assigned to it yet. The first block B_1 is initiated with l_1. B_1 will contain all trailers departing from the origin terminal o of l_1 after time zero and before the arrival of the first inbound trailer. All shipments that can be loaded on any of the trailers in B_1 are known, i.e., the shipments available at time zero at terminal o and any forecast shipments at terminal o before the end time of the block. Therefore, we can solve a (single) optimization problem that assigns these shipments to the trailers in B_1. Next, conceptually, we execute the trailer movements in B_1 and remove them. Thus, the shipments that were loaded into the trailers of B_1 are now available at the destination of these trailers at their time of arrival. After that, we proceed to create B_2 and so on. Note that this dynamic block

generation scheme implicitly assumes that the assignment of shipments to the trailers in a block is performed block by block in the order that the blocks are generated. Figure 2.5 shows the result of dynamic block generation for the same example as in Figure 2.4. Dynamic block generation has the advantage that it generates larger blocks compared to static block generation (six blocks compared to eight in the example). Algorithm 4 gives the pseudo-code for dynamic block generation.

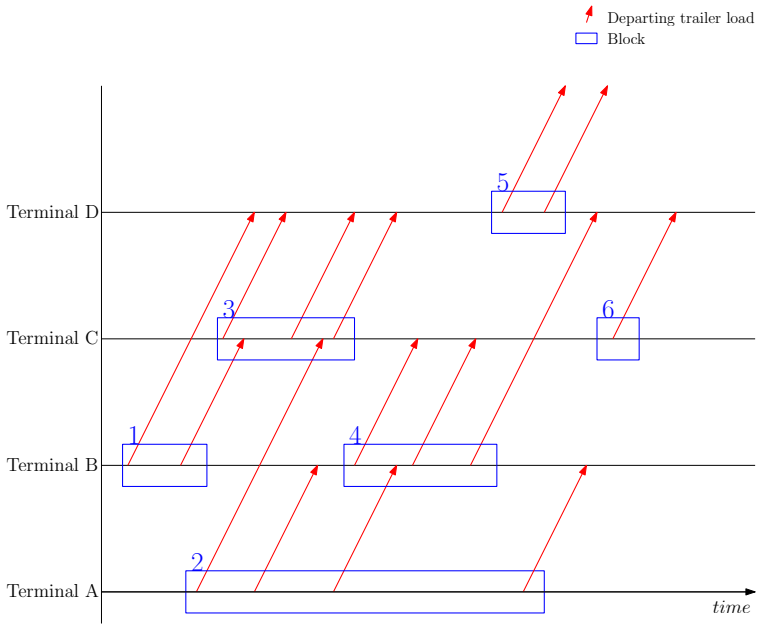


Figure 2.5: An example of the dynamic generation of blocks. The numbers at the top left corners of the blocks represents the order in which the blocks are created.

Assigning shipments to trailers

Recall that the planned flow and the planned trailer movements are designed to meet the service guarantees of the shipments in the system on a day with average demand. Therefore, the premise of all our approaches is that assigning shipments to trailers on their planned flow paths is desirable.

Therefore, all block loading approaches start by loading trailers with only planned path shipments and only after that, if there is remaining capacity, load alternate path shipments. The loading of the trailers in a block with planned path shipments proceeds as follows. For a given block $\mathcal{B}$, let $\mathcal{L}_{\mathcal{B}}$ be the set of trailers and let $\mathcal{S}_{\mathcal{B}}$ be the set of shipments available for loading. Furthermore,

Algorithm 4: Dynamic Block Generation

$\mathcal{B} \leftarrow \{\}$
$\bar{l} \leftarrow$ first trailer that dispatches in the network
$t_e \leftarrow 0$
$lastBlock \leftarrow false$
while $lastBlock$ *is* $false$ **do**
  $t_d, org \leftarrow$ dispatch time and origin of $\bar{l}$
  $\hat{l} \leftarrow$ first unassigned trailer that arrives at org at time $t > t_d$
  if $\hat{l} = \emptyset$ **then**
    $lastBlock \leftarrow true$
  else
    $t_e \leftarrow min\{$end of current sort, arrival time of $\hat{l}\}$
    $b \leftarrow$ list of all trailers that dispatch in the time window $[t_d, t_e)$ at terminal org
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{b\}$
    $\bar{l} \leftarrow$ first unassigned trailer that dispatches at time $t \geq t_d$
Return $\mathcal{B}$

let $\mathcal{L}_\mathcal{N}$ be the set of trailers departing after the end of the block but before the end of the sort. We load the trailers in $\mathcal{L}_\mathcal{B} \cup \mathcal{L}_\mathcal{N}$ in order of nondecreasing departure time with planned path shipments in $\mathcal{S}_\mathcal{B}$ in order of nonincreasing urgency, and, in case of ties, in order of nonincreasing size (see Algorithm 5 for details of procedure BLOCK-PF). The loading decisions for trailers in $\mathcal{L}_\mathcal{B}$ are fi-

Algorithm 5: BLOCK-PF

for ***each*** *trailer* $l \in \mathcal{L}_\mathcal{B}$ **do**
  $\mathcal{S}_l^P \leftarrow$ list of planned path shipments in $\mathcal{S}_\mathcal{B}$ available for pickup, sorted by urgency then by quantity, in descending order
  while *there's still capacity left in* l *or* $\mathcal{S}_l^P = \emptyset$ **do**
    load shipments from $\mathcal{S}_l^P$ in l
for ***each*** *trailer* $l \in \mathcal{L}_\mathcal{N}$ **do**
  $\mathcal{S}_l^P \leftarrow$ list of planned path shipments in $\mathcal{S}_\mathcal{B}$ available for pickup, sorted by urgency then by quantity, both in descending order **while** *there's still capacity left in* l *or* $\mathcal{S}_l^P = \emptyset$ **do**
    load shipments from $\mathcal{S}_l^P$ in l without carrying the decisions over to subsequent blocks

nal whereas the loading decisions for trailers in $\mathcal{L}_\mathcal{N}$ are tentative (as new shipments may become

available for loading in subsequent blocks, e.g., shipments with higher urgency). The reason for loading shipments in $\mathcal{S}_\mathcal{B}$ in trailers in $\mathcal{L}_\mathcal{N}$ is that we do not want to load too many shipments in $\mathcal{S}_\mathcal{B}$ on alternate paths, especially when these shipments can be loaded in trailers along their planned flow path later in the sort (but not in the block). Once shipments in $\mathcal{S}_\mathcal{B}$ have been loaded in trailers along their planned path, we have to decide whether any remaining capacity in the trailers in the block should be used to load any remaining shipments in $\mathcal{S}_\mathcal{B}$ in these trailers if they happen to be on their alternate path, or whether to postpone their loading to subsequent blocks. In the following, we will present different approaches for making these decisions.

A simple look ahead heuristic

Let $\mathcal{S}^u_\mathcal{B} \subseteq \mathcal{S}_\mathcal{B}$ be the set of as-yet unloaded shipments in nonincreasing order of urgency, and, in case of ties, in order of nonincreasing size. For each shipment $s \in \mathcal{S}^u_\mathcal{B}$, let $L^A_s \subseteq \mathcal{L}_\mathcal{B}$ be the set of trailers along the alternate path of s that have sufficient remaining capacity to load s. Furthermore, let

$$t^A_s = \min_{l \in L^A_s} EstArrival(s, dst_l, atm_l),$$

i.e., t^A_s is the earliest estimated arrival time of s at its destination if it is loaded in one of trailers in L^a_s, and let

$$l^A_s =_{l \in L^A_s} EstArrival(s, dst_l, atm_l).$$

Finally, let t^P_s be the estimated arrival time of s at its final destination if, instead, it is loaded in the first trailer departing in the next sort that is on its planned flow path. If $t^A_s < t^P_s$, we load s into l^A_s, otherwise, we do not load s in this block. That is, we only load a shipment on a trailer on its alternate path if the arrival at its destination is expected to be earlier than when loading is postponed until the next sort. Algorithm 6 gives the pseudo code of BLK-LOOKAHEAD.

BLK-LOOKAHEAD can also be implemented efficiently. The estimated arrival times at the destination for shipments $s \in \mathcal{S}_\mathcal{B}$ and trailers $l \in L^A_s$ are pre-computed and stored in a look-up

Algorithm 6: BLK-LOOKAHEAD

$\mathcal{S}^u_{\mathcal{B}} \leftarrow$ list of shipments in $\mathcal{S}_{\mathcal{B}}$, sorted by urgency, then by quantity in descending order
for *each* *shipment* s *in* $\mathcal{S}^u_{\mathcal{B}}$ **do**
  $L^A_s \leftarrow$ list of alternative path trailers available within the block with enough capacity left to load
  $l^P_s \leftarrow$ the first planned path trailer departing after the end of the sort containing the block
  $l^A_s \leftarrow_{l \in L^A_s} EstArrival(s, dst_l, atm_l)$
  $t^A_s \leftarrow$ estimated arrival of s at its final destination if loaded in l^A_s
  $t^P_s \leftarrow$ estimated arrival of s at its final destination if loaded in l^P_s
  if $t^A_s < t^P_s$ **then**
    load s in l^A_s
  else
    postpone the loading decision for s to the subsequent block

table using $EstArrival(s, dst_l, atm_l)$. The trailer in L^A_s resulting in the earliest estimated arrival time, l^A_s, can be determined at the same time at no extra cost.

Optimization: Basic formulation

Rather than deciding whether to assign a shipment to a trailer on its alternate path one shipment at a time, we next present an optimization model that decides whether to assign shipments to trailers on their alternate paths simultaneously. To allow postponing the loading of shipments to the next sort, which, at the same time, accommodates situations in which there is insufficient capacity to load all shipments on trailers on their alternate paths, we introduce a dummy trailer l^* with infinite capacity. For a given shipment $s \in \mathcal{S}^u_{\mathcal{B}}$, the feasible assignments, other than to l^*, are to trailers departing after the time that s becomes available and that are on the alternate path for s. We define the cost of assigning a shipment s to a trailer $l \in L^A_s$ by the function $C(s, l)$ given by

$$C(s, l) = \max\{EstArrival(dst_s, dst_l, atm_l) - due_s, 0\}.$$

The cost of assigning s to the dummy trailer l^* is set to $C(s, \bar{l})$, where $\bar{l}$ is the first trailer on the

planned path of s in the next sort or infinity if no such trailer exists. Recall that it is desirable to dispatch shipments in the sort in which they arrive, but that rolling over shipments is possible if it reduces the total lateness. We model the assignment of shipments to trailers within a block as the following integer program

$$\min \quad \sum_{s \in \mathcal{S}^u_{\mathcal{B}}} \sum_{l \in L(s) \cup l*} qty_s C(s,l) x_{sl}$$

$$s.t. \quad \sum_{l \in L^A_s \cup l^*} x_{sl} = 1 \qquad \forall s \in \mathcal{S}^u_{\mathcal{B}} \tag{2.8}$$

$$\sum_{\substack{s \in \mathcal{S}^u_{\mathcal{B}} \\ l \in L^A_s}} qty_s x_{sl} \leq cap_l \qquad \forall l \in \mathcal{L}_{\mathcal{B}} \tag{2.9}$$

$$x_{sl} \in \{0,1\} \qquad \forall s \in \mathcal{S}^u_{\mathcal{B}} \, \forall l \in L^A_s$$

where cap_l is the remaining capacity of trailer l and x_{sl} is the decision variable that models whether to assign s to trailer l ($x_{sl} = 1$) or not ($x_{sl} = 0$). The objective function seeks to minimize the lateness of the shipments in the block using estimates of a shipment's arrival time at its destination when it is loaded on a particular trailer in the block. The lateness of a shipment is weighted by its size, thus a higher priority is given to large shipments. Constraints (2.8) ensure that every shipment in the block is assigned to a trailer (possibly the dummy trailer l^*) and Constraints (2.9) ensure that the capacity of each trailer departing within the block is not exceeded.

Optimization: Extended formulation

The basic formulation seeks to minimize the lateness of the shipments in a block at a terminal, but ignores the impact that the loading of these shipments may have on the terminals where they end up and where they may not have been expected and where there may be insufficient capacity to handle them. To address this limitation, we propose an extended formulation that also considers shipments and capacity at the destinations of the trailers departing in the block, and, thus, estimates

the available capacity at these destinations to handle the shipments in the block. As a consequence, shipments are less likely to be send to destinations on their alternate paths if the available capacity at that destination is limited.

Let $\mathcal{L}_l^+$ be the set of trailers departing from the destination of l after its arrival there and before the end of the sort in which it arrives, and let $\mathcal{L}_{\mathcal{B}}^+ = \bigcup_{l \in \mathcal{L}_{\mathcal{B}}} \mathcal{L}_l^+$. Let $\mathcal{L}_s^+$ for $s \in \mathcal{S}_{\mathcal{B}}$ be the set of trailer pairs $\{l_1, l_2\}$ with $l_1 \in \mathcal{L}_{\mathcal{B}}$ a trailer on the alternate path for shipment s and $l_2 \in \mathcal{L}_{l_1}^+$ a trailer on the planned path for shipment s. Let $\mathcal{S}_{\mathcal{B}}^+$ be the set of shipments expected to be available at the destinations of trailers in $\mathcal{L}_{\mathcal{B}}$, which includes shipments that arrive there on their planned path and shipments that arrive there on one of their alternate paths from blocks that were processed previously. For each shipment $s \in \mathcal{S}_{\mathcal{B}}^+$, let $L_s^{\mathcal{P}}$ be the set of trailers departing after the time s becomes available, but before the end of the sort in which s becomes available, and that are on the planned path of s. We model the assignment problem of shipments to pairs of trailers as the following integer program:

$$\min \quad \sum_{s \in \mathcal{S}_{\mathcal{B}}^u} \sum_{\substack{\{l_1,l_2\} \in \\ \mathcal{L}_s^+ \cup \{l^*,l^*\}}} qty_s C_1(s, \{l_1, l_2\}) x_{s\{l_1,l_2\}} + \sum_{s \in \mathcal{S}_{\mathcal{B}}^+} \sum_{l \in L_s^{\mathcal{P}} \cup l^*} qty_s C_2(s, l) y_{sl}$$

$$s.t. \quad \sum_{\substack{\{l_1,l_2\} \in \\ \mathcal{L}_s^+ \cup \{l^*,l^*\}}} x_{s\{l_1,l_2\}} = 1 \qquad \forall s \in \mathcal{S}_{\mathcal{B}}^u \tag{2.10}$$

$$\sum_{l \in L_s^{\mathcal{P}} \cup l^*} y_{sl} = 1 \qquad \forall s \in \mathcal{S}_{\mathcal{B}}^+ \tag{2.11}$$

$$\sum_{\substack{s \in \mathcal{S}_{\mathcal{B}}^u: \\ \{l,l'\} \in \mathcal{L}_s^+}} qty_s x_{s\{l,l'\}} \leq cap_l \qquad \forall l \in \mathcal{L}_{\mathcal{B}} \tag{2.12}$$

$$\sum_{\substack{s \in \mathcal{S}_{\mathcal{B}}^u: \\ \{l',l\} \in \mathcal{L}_s^+}} qty_s x_{s\{l',l\}} + \sum_{\substack{s \in \mathcal{S}_{\mathcal{B}}^+: \\ l \in L_s^{\mathcal{P}}}} qty_s y_{sl} \leq cap_l \qquad \forall l \in \mathcal{L}_{\mathcal{B}}^+ \tag{2.13}$$

$$x_{s\{l_1,l_2\}} \in \{0,1\} \qquad \forall s \in \mathcal{S}_{\mathcal{B}}^u \; \forall \{l_1, l_2\} \in \mathcal{L}_s^+$$

$$y_{sl} \in \{0,1\} \qquad \forall s \in \mathcal{S}_{\mathcal{B}}^+ \; \forall l \in L_s^{\mathcal{P}}$$

In this extended formulation, we not only consider the capacity of trailers in the block, i.e., Constraints (2.12), but also the capacity of the outbound trailers at the destinations of the trailers in the block, i.e., Constraints (2.13). This ensures that we load shipments in trailers on their alternate paths only if their is likely sufficient capacity at the destinations of these trailers. As in the basic formulation we allow postponing shipments through the introduction of dummy trailer pairs $\{l^*, l^*\}$ and $\{l, l^*\}$ for shipments in $\mathcal{S}_{\mathcal{B}}$ and dummy trailer l^* for shipments in $\mathcal{S}_{\mathcal{B}}^+$. The cost of assigning a shipment s in $\mathcal{S}_{\mathcal{B}}$ to a trailer pair $\{l_1, l_2\} \in \mathcal{L}_s^+$ is given by:

$$C_1(s, \{l_1, l_2\}) = \begin{cases} C(s, \bar{l}), \text{ if } \{l_1, l_2\} = \{l^*, l^*\} \\ C(s, \hat{l}), \text{ if } l_1 \neq l^* \text{ and } l_2 = l^* \\ C(s, l_2), \text{ o.w} \end{cases}$$

and the cost of assigning a shipment s in $\mathcal{S}_{\mathcal{B}}^+$ to a trailer $l_2 \in L_s^{\mathcal{P}}$ is given by:

$$C_2(s, l_2) = \begin{cases} C(s, \hat{l}), \text{ if } l_2 = l^* \\ C(s, l_2), \text{ o.w} \end{cases}$$

where $\bar{l}$ is the first planned path trailer leaving after the end of the current sort containing the block and $\hat{l}$ is the first planned path trailer leaving after the end of the sort that contains the trailer l_2. The set of forecast shipments in $\mathcal{S}_{\mathcal{B}}^+$ is obtained as follows: before creating and solving blocks, we (tentatively) load the trailers in the network with planned path shipments using URG-PULL-PF. The resulting loading decisions are only used for estimating the remaining capacity of trailers. After solving a block using the extended formulation, the new location and arrival time for a shipment assigned to pair $\{l_1, l_2\}$ are set to dst_{l_1} and atm_{l_1}, respectively.

Block Loading

Once the blocks have been created, they are sorted in order of nonincreasing departure time of the first trailer in the block, and the blocks are processed one after the other. The construction of the blocks and the processing order guarantee that all shipments that can be loaded into the trailers of a block are known at the start time of the block. For blocks that contain a large number of shipments and trailers, the optimization models may become too large to be solved in an acceptable amount of time. In order to prevent spending too much time on a single block, we set a time limit for solving the integer program. If the optimal solution is not obtained within the time limit, we check if the gap between the upper and lower bound is less than a given threshold (discussed in Section 4.6). If so, then we accept the incumbent solution and move on to the next block. Otherwise, we solve the block using the look ahead loading strategy for blocks. Algorithm 7 shows the overall procedure for a block loading heuristic.

Algorithm 7: Block Loading

$Blocks \leftarrow$ GENERATEBLOCKS(); *using either static or dynamic generation*

for ***each*** *block* ***b*** *in* $Blocks$*, sorted by the dispatch time of the first trailer departing within the block* **do**

  BLOCK-PF(b); *Solves the assignment problem for planned path trailers*

  SOLVEBLOCK(b); *Solves the assignment problem for the remaining shipments in the block using either lookahead heuristic or the basic/extended formulation* **if** *optimization models are used, timeLimit is exceeded and optimality gap* $> p$ **then**

    Blk-LookAhead(b); *Solves the assignment problem for the remaining shipments using the look ahead loading heuristic for blocks*

2.4 Computational Experiments

2.4.1 Performance metrics

In order to assess the quality of the different loading strategies, we use the following set of performance metrics, where we only take into account shipments that enter the system in the first 24 hours of the planning period. These shipments matter most as they have been and still are in the system or are just entering the system and they are directly impacted by the decisions we make (at time zero). For completeness sake, we present results for all shipments, i.e., including forecast shipments in Tables A.1 and A.2 in the appendix.

- **Total Lateness (TL)**. For a given shipment s, we define the lateness $late_s$ as the difference between the arrival time of the shipment at its destination and its due time. There are two cases to consider: (i) the shipment arrives at its destination during the planning period, and (ii) the shipment does not arrive at its destination during the planning period. In the second case, the arrival time of the shipment is estimated based on the use of the earliest departing trailers along the shipment's planned path after the planning period. If a shipment arrives on time (before its due time), $late_s$ is set to zero as we are only interested in the shipments that arrive late. The total lateness for a loading strategy is then defined as the sum of each shipment's lateness weighted by its size:

$$TL = \frac{1}{\sum_{s \in S} qty_s} \sum_{s \in S} qty_s \, late_s$$

  We define TL_{24} as the total lateness of shipments that enter the system in the first 24 hours of the planning period and that have a due time in the planning period.

- **Velocity at time t (V_t)**. For a given shipment s available for pickup at location w and time t, we define *velocity* as the ratio of the expected transit time from origin to destination and

the available transit time from origin to destination:

$$V_t^{s,w} = \frac{EstArrival(s,w,t) - otm_s}{due_s - otm_s},$$

When location w is the shipment's destination, we replace the estimated arrival time by the actual arrival time. The velocity of the network at time t is then defined as the average velocity of the shipments weighted by size:

$$V_t = \frac{1}{\sum_{s \in S} qty_s} \sum_{s \in S} qty_s \, V_t^{s,w},$$

In order to compute the velocity at times 0, 24, and 48, we record the following information for each shipment:

- $V_0^{s,w}$: the velocity at the terminal where the shipment enters the network at the time it enters the network.
- $V_{24}^{s,w}$: the velocity at the first terminal the shipment visits at or after time 24 at the time of arrival at that terminal, or, if the time of arrival is before 24, at 24.
- $V_{48}^{s,w}$: the velocity at the first terminal the shipments visits at or after time 48 at the time of arrival at that terminal, or, if the time of arrival is before 48, at 48.

The velocity of the system gives an indication of the likelihood that the shipments in the system will be delivered on time (a low value is better than a high value). The purpose of the velocity values is mostly to recognize changes in the system, i.e., if the velocity of the system increases, then the system is getting under more pressure and the risk of late deliveries increases.

- **%D**, $\mathbf{\%D_{OT}}$, and $\mathbf{\%D_{L}}$: the fraction of shipments delivered, the fraction of shipments delivered or expected to be delivered on time, and the fraction of shipments delivered late or

expected to be delivered late.

- **%ND** and **%NDL**: the fraction of shipments that did not reach their destination during the planning period and the fraction of these shipments that are known to reach their destination late.

- **%PF** and **%AF**: the fraction of shipments that only used trailers on their planned path during the planning period and the fraction of shipments that used at least one trailer on their alternate path during the planning period.

- **%S**: the fraction of shipments that have a due time within the planning period and that have "stalled", i.e., have not been loaded into any trailer during the planning period.

- **RO-AVG**: the average number of sorts that shipments were rolled over during planning period weighted by size; let I^s bet the set of terminals visited by shipment s during the planning period and let nst_i^s be the number of sorts used by shipment s at terminal $i \in I^s$, then

$$\text{RO-AVG} = \frac{1}{\sum_{s \in \mathcal{S}_{24}} qty_s} \sum_{s \in \mathcal{S}_{24}} qty_s \frac{\sum_{i \in I^s} nst_i^s}{|I^s|}$$

2.4.2 Instances

The set of ten instances used in the computational experiments are derived from snapshots of historical data from a major U.S. LTL carrier. Each snapshot corresponds to a set of consecutive days at different times of the year. The number of shipments varies, but the planned and alternate paths as well as the trailer movements are similar. Each instance contains shipment and trailer information from more than 72 continuous hours of operations. We set the start of the planning period to be 52 hours before the last departing trailer so as to ensure that we have information on all trailers departing during the 48 hour planning period. Given that each snapshot represents

historical real-world data with a considerable number of imperfections, we have decided to make the following modifications:

- Shipments meeting at least one of the following criteria are removed from the instance: (a) shipments with the same origin and destination, (b) shipments with an incomplete planned path, (c) shipments with no trailer departing along their planned path at the terminal where they enter the network; (d) shipments with a size larger than the capacity of a trailer; (e) shipments that are already late at time zero; (f) shipments that have been in the system for more than 96 hours before the start of the planning period;

- The due time of a shipment s that will arrive late at its destination if loaded in the earliest possible departing trailers along the planned path is changed as follows. We estimate the arrival time atm_s of the shipment at its destination given its origin location and the time it entered the network by always considering the earliest departing trailers in the planned path and set the due time due_s as:

$$due_s = atm_s + 4$$

These changes ensure that each shipment can reach its destination before its due time if it would be the only shipment in the system. Table 4.1 shows the number of shipments ($|S|$), the number of trailers ($|L|$) and the velocity of the network at the start of the planning period (V_0) for each instance. The instances are based on a network with about 350 terminals. A terminal can operate up to four sorts (*Day*, *Twilight*, *Night*, and *Sunrise*). About 85% of the terminals, mostly End-of-Lines, operate two sorts, about 10%, operate three sorts, and about 5%, mostly Breakbulks, operate four sorts.

Table 2.1: Information on the instances used in the computational experiments.

Instance	$\|L\|$	$\|S\|$	V_0
I1	12,134	92,329	0.70
I2	12,072	73,608	0.69
I3	12,105	77,290	0.70
I4	10,953	71,592	0.74
I5	11,966	91,747	0.63
I6	11,677	72,001	0.63
I7	11,723	89,617	0.64
I8	11,911	91,391	0.63
I9	11,806	88,937	0.64
I10	11,806	89,619	0.65

2.4.3 Analysis

We compare the performance of the following loading strategies: FIFO-PUSH, URG-PULL, URG-PULL-PF, i.e., the variant of URG-PULL in which shipments are only loaded on trailers along their planning path, BLK-LOOKAHEAD, block loading with look ahead, BLK-IP-BASIC, block loading using the basic assignment formulation, and BLK-IP-EXTENDED, block loading using the extended assignment formulation. All block loading variants use dynamic generation of blocks. When solving an integer program in one of the formulation-based block loading variants, a time limit of 300 seconds is imposed and solutions with an integrality gap of less than 10% are considered acceptable.

All loading strategies are coded in C# and integer programs are solved using IBM CPLEX Optimizer 12.6. All experiments were conducted in a single thread of a dedicated Intel Core i5-73000U 2.60GHz CPU with 16GB RAM running Microsoft Windows 10.

The results can be found in Table 2.2.

We see that using alternate paths too aggressively, as is done in URG-PULL, results in poor performance. However, the results also show that using alternate paths too sparingly, as is done in URG-PULL-PF, which does not use alternate paths at all, does not result in strong performance

Table 2.2: Results for the set of instances used in the computational experiments considering different metrics. The best results for each instance in terms of TL, $\%D$, $\%D_{\mathrm{OT}}$, and **RO-AVG** are highlighted in bold. TL is given in hours, and total runtime **TT** is in seconds.

Ins.	Algorithm	TL	$\%D$	$\%D_{\mathrm{OT}}$	$\%D_{\mathrm{L}}$	%PF	%AF	$\%S$	RO-AVG	TT
	FIFO-Push	6.02	87.53	63.67	23.87	77.87	21.51	0.69	1.03	535.01
	Urg-Pull	3.73	92.01	73.03	18.98	82.96	16.09	0.56	0.93	424.19
I1	Urg-Pull-PF	3.66	92.44	72.73	19.71	97.78	0.00	1.12	1.11	170.36
	Blk-LookAhead	3.13	93.29	75.00	18.29	88.23	10.82	0.52	0.70	515.94
	Blk-IP-Basic	**3.06**	**93.48**	75.36	18.12	91.29	7.71	0.50	0.67	829.54
	Blk-IP-Extended	**3.06**	**93.48**	**75.45**	18.03	91.96	7.00	0.51	**0.57**	8,468.78
	FIFO-Push	3.61	90.43	74.18	16.24	70.56	25.64	3.63	0.79	370.37
	Urg-Pull	2.46	93.6	79.74	13.87	73.87	21.05	3.29	0.81	290.67
I2	Urg-Pull-PF	2.15	94.02	81.13	12.89	92.12	0.00	4.20	0.98	130.72
	Blk-LookAhead	1.64	95.11	83.57	11.54	82.90	12.07	3.34	0.59	366.13
	Blk-IP-Basic	1.58	95.13	83.86	11.27	88.29	5.83	3.29	0.61	550.80
	Blk-IP-Extended	**1.56**	**95.24**	**84.03**	11.21	88.38	5.73	3.31	**0.54**	5,515.14
	FIFO-Push	3.83	89.3	73.56	15.74	71.33	24.47	3.81	0.74	363.97
	Urg-Pull	2.49	93.24	79.80	13.43	75.59	20.08	2.74	0.73	323.12
I3	Urg-Pull-PF	2.19	93.61	80.84	12.77	92.82	0.00	3.52	0.88	140.40
	Blk-LookAhead	1.78	94.45	83.08	11.37	83.78	11.88	2.62	0.46	369.80
	Blk-IP-Basic	1.74	94.65	83.36	11.30	89.16	5.48	2.51	0.48	570.22
	Blk-IP-Extended	**1.73**	**94.67**	**83.45**	11.22	89.30	5.35	2.60	**0.42**	5,842.59
	FIFO-Push	5.21	88.51	70.09	18.42	74.95	21.87	3.10	0.71	324.61
	Urg-Pull	3.62	92.28	76.04	16.24	76.28	20.07	3.03	0.74	256.43
I4	Urg-Pull-PF	3.30	92.49	77.27	15.22	93.99	0.00	4.22	0.89	117.97
	Blk-LookAhead	2.44	94.06	79.91	14.15	85.11	11.35	2.86	0.61	319.58
	Blk-IP-Basic	2.39	94.21	80.32	13.89	89.69	6.48	2.86	0.60	467.32
	Blk-IP-Extended	**2.37**	**94.24**	**80.34**	13.90	90.03	6.12	2.95	**0.55**	5,531.15
	FIFO-Push	3.42	94.55	77.99	16.56	78.43	20.27	1.72	0.99	404.95
	Urg-Pull	2.58	95.83	82.99	12.84	76.61	21.36	1.50	0.99	284.41
I5	Urg-Pull-PF	1.84	97.06	86.40	10.67	96.79	0.00	1.64	1.15	127.23
	Blk-LookAhead	1.61	97.44	87.55	9.89	87.6	10.48	1.5	**0.86**	368.22
	Blk-IP-Basic	1.54	97.67	**88.01**	9.66	93.52	4.11	1.35	0.90	519.31
	Blk-IP-Extended	**1.51**	**97.71**	87.96	9.75	93.91	3.69	1.43	**0.86**	5,372.42
	FIFO-Push	4.11	96.16	73.85	22.31	80.5	17.93	1.79	1.62	305.79
	Urg-Pull	3.31	96.87	77.62	19.25	78.71	19.05	1.56	**1.45**	205.03
I6	Urg-Pull-PF	2.75	98.05	81.84	16.21	97.12	0.00	1.80	1.66	113.14
	Blk-LookAhead	2.69	98.24	81.94	16.30	86.74	11.10	1.47	1.47	299.84
	Blk-IP-Basic	2.63	98.29	82.18	16.11	94.14	3.49	1.44	1.52	410.41
	Blk-IP-Extended	**2.61**	**98.37**	**82.33**	16.04	94.55	2.98	1.59	1.50	3,601.03
	FIFO-Push	3.19	94.53	78.82	15.71	78.39	20.34	1.67	0.90	409.99
	Urg-Pull	2.39	96.03	83.21	12.82	76.00	21.82	1.43	0.90	285.85
I7	Urg-Pull-PF	1.63	97.64	86.83	10.80	96.88	0.00	1.49	1.05	137.23
	Blk-LookAhead	1.36	97.94	88.04	9.90	87.49	10.57	1.27	**0.86**	373.42
	Blk-IP-Basic	1.33	98.08	**88.22**	9.87	93.92	3.57	1.29	0.90	530.14
	Blk-IP-Extended	**1.31**	**98.09**	**88.22**	9.88	94.24	3.30	1.32	**0.86**	5,004.31
	FIFO-Push	3.12	94.88	78.98	15.90	78.83	20.10	1.32	0.88	421.18
	Urg-Pull	2.46	96.27	83.11	13.17	75.74	22.25	1.24	0.86	299.62
I8	Urg-Pull-PF	1.82	97.69	86.51	11.18	97.03	0.00	1.33	1.01	133.76
	Blk-LookAhead	1.67	97.95	87.46	10.49	87.57	10.83	0.99	**0.80**	383.82
	Blk-IP-Basic	1.63	98.01	87.72	10.29	94.03	3.59	1.02	0.84	554.16
	Blk-IP-Extended	**1.59**	**98.07**	**87.80**	10.27	94.33	3.34	1.02	0.81	5,318.20
	FIFO-Push	3.09	95.02	79.25	15.78	78.61	20.31	1.53	0.96	416.13
	Urg-Pull	2.42	96.24	83.23	13.01	75.93	22.25	1.40	0.87	290.05
I9	Urg-Pull-PF	1.85	97.55	86.24	11.31	97.18	0.00	1.76	1.04	127.76
	Blk-LookAhead	1.66	97.89	87.32	10.57	87.25	11.22	1.27	0.87	385.93
	Blk-IP-Basic	**1.60**	97.95	**87.57**	10.38	94.27	3.73	1.19	0.90	559.31
	Blk-IP-Extended	**1.60**	**97.96**	87.49	10.46	94.38	3.59	1.37	**0.87**	5,181.67
	FIFO-Push	3.46	94.67	76.25	18.42	78.95	19.73	1.57	1.04	424.19
	Urg-Pull	2.78	95.74	79.79	15.96	76.11	21.87	1.66	0.93	297.53
I10	Urg-Pull-PF	2.29	97.07	82.93	14.14	97.20	0.00	1.76	1.09	137.76
	Blk-LookAhead	2.17	97.40	83.87	13.53	87.18	11.07	1.33	**0.85**	395.01
	Blk-IP-Basic	2.10	97.50	84.19	13.30	94.06	3.80	1.29	0.88	560.56
	Blk-IP-Extended	**2.08**	**97.57**	**84.24**	13.33	94.24	3.57	1.47	**0.85**	5,391.10

either (although clearly better). Furthermore, the results for URG-PULL-PF show that not using alternate paths leads to a considerable increase in the number of stalled shipments, and, more general, a higher number of roll-overs.

All block loading strategies, which seek to balance the use of planned flow and alternate flow paths, perform noticeably better. For BLK-LOOKAHEAD, we see an average improvement of 48.41% in total lateness over the total lateness of FIFO-PUSH, 29.54% over the total lateness of URG-PULL, and 13.79% over the total lateness of URG-PULL-PF (with largest improvements of 57.36% and 43.09% for Instance 7, and 26.06% for Instance 4, respectively). We also see higher fractions of delivered shipments.

In all block loading strategies, the trailers are loaded first with shipments for which the trailers are on their planned flow path, and the same algorithm, BLOCK-PF, is used for all block loading strategies. Therefore, the difference is a result of the choice of alternative paths. Block loading strategies are conservative in their use of alternative paths: they are only used when a shipment is expected to arrive earlier at its destination compared to waiting for the next planned path trailer. This accounts for the improvements not only in velocity and total lateness, but also in number of rolled over shipments.

Not surprisingly, using optimization models to refine the assignment of shipments to trailers improves performance. Using BLK-IP-BASIC, we see an average improvement of 2.82% in total lateness over BLK-LOOKAHEAD (with largest improvement of 4.34% for Instance 5). BLK-IP-EXTENDED performs even better, which shows the importance of evaluating the impact of sending shipments to an alternative destination. By taking into account the available capacity in the planned path trailers at an alternative destination, better loading decisions are made. More precisely, resorting to alternative paths is only allowed in case there is sufficient capacity in the planned path trailers at the destination of an alternative path trailer. As a result of this careful examination of alternative paths, more shipments are rolled over and pushed into their planned path trailers in subsequent sorts. This explains the improvement in planned flow percentage of BLK-IP-EXTENDED over

BLK-IP-BASIC. However, note that the average number of sorts used per shipment in BLK-IP-EXTENDED is still comparable to BLK-IP-BASIC. This is explained by the fact that we are making loading decisions that give priority to rolling over shipments to subsequent sorts where they can be potentially loaded in a planned path trailer, rather than sending them to an alternative destination with no immediate available capacity. When compared to URG-PULL-PF, BLK-IP-EXTENDED shows an average improvement of 17.10% in total lateness (with a largest improvement of 28.18% for Instance 4). It does better in most other metrics as well. However, the improvement in performance of BLK-IP-EXTENDED comes at a price. The computing time increases by a factor of 10, on average, compared to BLK-IP-BASIC.

Next, we investigate the formulation-based block loading strategies in some more detail. Table 2.3 present statistics on the blocks generated for both the basic and the extended formulations. We report the number generated ($\#B$), the average duration (D_{Avg}), the average number of trailers (L_{Avg}), the average number of pairs of trailers in the extended formulation (P_{Avg}), the average number of shipments using the basic and the extended formulations (S^B_{Avg} and S^E_{Avg}, respectively), the average solution time for the basic and extended formulations (T^B_{Avg} and T^E_{Avg}, respectively), and the maximum IP solve time for the basic and extended formulations (T^B_{Max} and T^E_{Max}, respectively). We see that the average number of trailers per block is more than ten for all the

Table 2.3: Statistics on the dynamic generation of blocks for each instance.

Instance	$\#B$	D_{Avg}	L_{Avg}	S^B_{Avg}	T^B_{Avg}	T^B_{Max}	P_{Avg}	S^E_{Avg}	T^E_{Avg}	T^E_{Max}
I1	949	2.06	13.79	123.47	3.08	74.22	211.11	123.39	14.48	176.61
I2	953	2.09	13.67	71.64	2.05	41.05	210.91	71.75	9.02	150.97
I3	953	2.07	13.70	77.10	2.09	39.88	209.29	77.18	9.77	119.39
I4	859	2.13	13.75	69.17	2.03	44.25	228.11	69.26	11.00	195.46
I5	926	2.11	13.92	77.58	2.35	49.71	224.19	77.91	12.08	157.79
I6	914	2.11	13.78	67.04	1.80	23.82	224.49	67.24	8.69	89.08
I7	921	2.07	13.73	77.39	2.40	47.37	218.85	77.98	11.29	134.95
I8	925	2.06	13.88	80.47	2.35	40.40	223.06	81.09	11.87	172.02
I9	923	2.05	13.79	79.72	2.42	38.14	220.93	80.10	11.70	146.47
I10	919	2.05	13.87	88.87	2.55	33.78	221.22	89.28	12.11	136.16

instances. This explains, to some extent, why the block-based loading strategies, and, thus, the formulation-based loading strategies, perform better than trailer-based loading strategies. We also see that the solution time for the integer programs solved when using BLK-IP-EXTENDED is, on average, about five times more than when using BLK-IP-BASIC.

Interestingly, looking at the historic decisions, in two days of operations, 19.3% of shipments were loaded on a trailer that was not on the planned or on an alternative path in some point of their journey. This is likely due to the fact the set of alternate paths provided to us was not up to date and that terminal managers often make decisions based on their own judgment and experience, rather than following guidelines. Furthermore, 3.5% of the trailers in the network had their capacity constraint violated by 20% or more. This is mostly due to the imprecision in estimating the size (volume and weight) of a shipment. This shows that introducing decision support in near real-time load adjustments is challenging...

2.5 Final remarks

We have designed and implemented heuristics that can be used for near real-time loadplan adjustments. Consolidation carriers have long recognized the opportunity and value of near real-time loadplan adjustments, but have also acknowledged the challenges of doing so in practice. These challenges relate to the data needs and the computational requirements. Detailed information on the system status, e.g., what pallets have already been loaded into a truck at a loading dock at a breakbulk terminal, is not always readily available, and to be able to react quickly to observed changes into anticipated freight volumes decision support tools have to be efficient, e.g., propose loadplan adjustments in 15 minutes or less. With the advances in data collection technology, the advances in computing power, and the advances in algorithms, we have reached the point where near real-time loadplan adjustments are possible. The heuristics described in this paper are in daily use at a large national US LTL carrier and are generated significant benefits.

The next phase of this research is to extend the technology to offer additional functionality:

adding or canceling schedules. For example, under-utilized trailers and schedules can be detected and the heuristics can be used to evaluate whether the shipments in these trailers and can be rerouted on alternate paths, and, if so, the schedules can be canceled. Or, when too many shipments stall at a terminal during a sort, the heuristics can be used to evaluate whether adding a schedule (one or more trailers) results in significantly fewer shipments stalling. Such functionality would further enhance a carrier's ability to better manage daily operational costs while maintaining the service guarantee promised to its customers.

CHAPTER 3

SUBSTITUTION-BASED EQUIPMENT BALANCING IN SERVICE NETWORKS WITH MULTIPLE EQUIPMENT TYPES

3.1 Introduction

Package express companies, such as FedEx and United Parcel Service, use a large and heterogeneous pool of trailers and containers in their service (linehaul) networks. A major challenge in the planning process is to ensure that the right equipment is available at the right location at the right time. This is difficult to achieve, in part, because the flow of packages between facilities in the network is not balanced. As a consequence, the companies are forced to move equipment empty, i.e., reposition equipment, which is expensive.

To reduce the complexity of their planning process, a large package express carrier typically applies a phased approach. In a flow planning phase, a forecast of daily origin-destination demand is used to determine origin-destination paths for packages that guarantee that service commitments are met and that create consolidation opportunities (consolidation is the primary mechanism a package express carrier employs to reduce/control its operational costs). In a load planning phase, the package flows are converted into loads, i.e., timed movements of equipment through the network. This phase continues to focus primarily on the flow of packages (now in discrete units - by assigning the flows of packages to equipment types), but equipment repositioning decisions are also made. In a scheduling phase, driver schedules are created to actually move the loads from their origin to their destination directly or through one or multiple relay points. A driver schedule plan typically covers a period of a week and has to satisfy many requirements, e.g., Hours of Service regulations and union contract rules. As considerations related to the equipment pool are only of secondary importance in the above planning phases, the resulting plan (i.e., the plan of loads to be

moved in the coming week and the driver schedules to execute this plan) is typically imbalanced, in the sense that the inventory of the different equipment types at a facility at the start of the week differs from that at the end of the week, which is referred to as the equipment imbalance introduced by (or associated with) the plan. As this may lead to a surplus or a shortage of equipment in the future, the final phase in the planning process seeks to reduce the equipment imbalance introduced by the plan. This last phase is the topic of this chapter. More specifically, we try to decrease the imbalance introduced by a plan by substituting the equipment types assigned to the loads in the plan. Equipment substitution complements empty repositioning of equipment, but is only possible if companies operate multiple, exchangeable equipment types. Equipment substitution has the advantage (over empty repositioning) that it does not incur any costs. Equipment substitution can implicitly introduce empty repositioning if equipment can be, and is, assigned to scheduled bobtail movements in the plan. (A bobtail movement is one in which a driver drives a tractor without any trailers.)

The existing literature on equipment management in the trucking industry focuses on the design of empty repositioning strategies to balance equipment (also referred to as "empty vehicle allocation" or "redistribution"), e.g., [32], [33], [34], and [35]. We are not aware of any literature on approaches based on equipment substitution to deal with equipment imbalance in the trucking industry. [36] present an overview of empty fleet management issues and strategies and introduce a taxonomy of empty flow problems related to the distribution and scheduling of empty movements. The strategies for empty equipment redistribution can be classified into two groups: (i) decentralized models where one facility operates and controls its own fleet and seeks to optimize its own performance metrics, and (ii) centralized models where an entire service network is considered and decisions are made that seek to optimize a set of global performance metrics. An example in the first group is [32], which proposes a decentralized stock control policy approach for both fleet sizing and empty repositioning restricted to a given center-terminal system (such system is comprised of one center and a group of terminals connected to it). An example in the second group is [33],

which proposes, in the context of maritime logistics, a mixed integer programming formulation to minimize the cost of repositioning empty containers in a region by finding the optimal locations for inland depots. Demand uncertainty, i.e., uncertainty of anticipated future freight flows, greatly affects empty repositioning. Demand uncertainty is typically approached using robust or stochastic optimization. [34], for example, propose a robust recovery optimization framework that can be applied to empty repositioning problems. [35] introduce a two-stage stochastic programming model for an environment with uncertain demand for and supply of containers, in which the objective is to minimize the costs of repositioning containers empty. They propose a sample average approximation method using a progressive hedging heuristic. In a maritime logistics context, [37] formulate an empty container substitution problem to minimize the cost of transporting empty containers. In their model, substitutions are allowed between container types based on their intended use, dimensions, and ownership.

In their ground networks, package express carriers employ many different types of equipment, or trailers, which are grouped into categories based on their size, i.e., *shorts* (trailers with a length of 28 feet), *longs* (trailers with a length ranging from 40 to 48 feet), and *extra longs* (trailers with a length of 53 feet). Equipment types can be combined into composite types. For example, a common composite type is a combination of two shorts. Other composite types are three shorts, a long combined with a short, etc.. These are allowed only in certain states. In general, an equipment type assigned to a load can be substituted by a larger equipment type as long as the origin and destination facility of the load can accommodate the larger equipment type. In some situations, an equipment type assigned to a load can be substituted by a smaller equipment type, but only if the capacity of the smaller equipment type is sufficient to accommodate the original load quantity.

As mentioned above, a plan for the coming week, i.e., the loads to be moved and the driver schedules to make this happen, may result in a change in the inventory of an equipment type at a facility at the end of the week. This happens when, in the plan, the number of loads departing from the facility with a specific equipment type is different from the number of loads arriving at

the facility with that equipment type. In this case, we say that the plan is imbalanced. We define the imbalance (induced by a plan) of a facility to be the sum of the imbalances (surplus or deficit) of the equipment types at the facility, and the total imbalance (induced by a plan) as the sum of the imbalances of the facilities in the network. The primary goal of substitution-based equipment balancing is to minimize the total imbalance induced by a plan with the least empty repositioning cost. A secondary goal is to achieve the minimum total imbalance with as few equipment substitutions as possible. That is, substitution-based equipment balancing is a hierarchical optimization problem.

The main contributions of our research are as follows:

- We introduce a staged approach to solve the substitution-based equipment balancing problem where we first minimize imbalance by means of equipment substitutions, then by means of empty repositioning. This approach mimics and optimizes current industry practice,
- We explore the value of combining equipment substitution decisions and empty repositioning in a single integrated model,
- We present two simple, but effective decomposition heuristics that yield high-quality solutions in a short amount of time,
- We conduct a computational study, using real-world instances, to assess the efficacy of our solution approaches and to analyze the benefits of substitution-based equipment balancing.

The remainder of the chapter is organized as follows. Section 3.2 introduces notation and presents integer programming formulations of the staged approach. Section 3.3 addresses the integrated model where equipment substitution decisions are combined with empty repositioning decisions. Section 3.4 discusses the computational study.

3.2 Staged Approach

In this section, we present a staged approach to solving the equipment balancing problem by decoupling equipment substitution and empty repositioning decisions. In Stage 1, we minimize equipment imbalance by means of equipment substitutions only. This stage is solved hierarchically in two phases. In a primary phase, the objective is to minimize the total equipment imbalance in the network. In a secondary phase, we minimize the number of equipment substitutions required to achieve the minimal imbalance determined in Phase 1. In Stage 2, we address the remaining imbalance in the network by empty repositioning decisions, i.e., introducing additional empty loads in the original load plan to reach zero-imbalance. We assume throughout the paper that it is always possible to reach zero-imbalance by empty repositioning. For that, the service network needs to satisfy the necessary and sufficient condition presented in Appendix C.

3.2.1 Notation and Formulations

We first introduce the notation used throughout the paper. A network N is represented as a directed graph, $N = (V, A)$, with each vertex representing a facility and each arc representing a load. A load represents a movement of equipment that is scheduled to dispatch during the planning horizon and deliver a quantity of packages for an origin-destination pair. It is also also characterized by the initial equipment type assigned to it which is used to compute the initial imbalance. Let $\mathbb{Z}_{\geq 0}$ be the set of non-negative integers, $n = |V|$ be the number of vertices, and $m = |A|$ be the number of arcs. Let $\mathcal{E}$ be the set of basic equipment types and $\mathcal{C}$ be the set of equipment type configurations used operationally, which can be a basic type (single unit of equipment) or a composite type (combination of multiple units of equipment) formed by combining basic types. For $c \in \mathcal{C}$, we have $c = \sum_{e \in \mathcal{E}} f_{ce} e$ with $f_{ce} \in \mathbb{Z}_{\geq 0}$ indicating how many units of basic equipment type $e \in \mathcal{E}$ are used in the composite type c. For each $v \in V$, let $\mathcal{C}_v \subseteq \mathcal{C}$ be the set of allowable equipment types at vertex v, and $\delta_v^+ = \{(v, u) \in A : u \in V\}$ and $\delta_v^- = \{(u, v) \in A : u \in V\}$

be the sets of outgoing and incoming arcs at v, respectively. Let $\sigma_v^+ := |\delta_v^+|$ and $\sigma_v^- := |\delta_v^-|$. An equipment assignment (or assignment for short) is a function $\mathscr{A} : A \to \mathcal{C}$ that assigns an equipment type to each arc. The initial assignment is denoted by $\mathscr{A}_0$. Let $\mathcal{C}_a \subseteq \mathcal{C}$ be the set of equipment types that can be assigned to arc $a \in A$, $A_{vc}^+ := \{a \in \delta_v^+ : c \in \mathcal{C}_a\}$, and $A_{vc}^- := \{a \in \delta_v^- : c \in \mathcal{C}_a\}$. For a given network N, let I^* be the minimum imbalance and $I(\mathscr{A})$ be the imbalance of an assignment $\mathscr{A}$. When $I(\mathscr{A}) = I^*$, we say $\mathscr{A}$ is optimal for N.

In Stage 1, our goal is to find an optimal $\mathscr{A}^*$ that is closest to $\mathscr{A}_0$, i.e., $\mathscr{A}^* \in \text{argmin}_{I(\mathscr{A})=I^*} \|\mathscr{A} - \mathscr{A}_0\|$, where $\|\mathscr{A} - \mathscr{A}_0\| = |\{a \in A : \mathscr{A}(a) \neq \mathscr{A}_0(a)\}|$. We use a two-phase hierarchical optimization approach, where we compute I^* for network N in Phase 1, and find the desired $\mathscr{A}^*$ in Phase 2. in Stage 2, we address the remaining imbalance I^* by means of empty repositioning.

3.2.2 Stage 1: Minimizing imbalance with the least equipment substitutions

Phase 1: Minimizing imbalance

We define the following variables. For $a \in A$ and $c \in \mathcal{C}_a$, let

$$y_{ac} = \begin{cases} 1, & \text{if equipment } c \text{ is used on arc } a, \\ 0, & \text{otherwise.} \end{cases}$$

For $v \in V$, $e \in \mathcal{C}_v \cap \mathcal{E}$, let $r_{ve} \in \mathbb{Z}_{\geq 0}$ be the imbalance for basic equipment type e at vertex v. The following model minimizes the total imbalance I^*:

$$I^* = \min \quad \sum_{v \in V} \sum_{e \in \mathcal{C}_v \cap \mathcal{E}} r_{ve} \tag{3.1}$$

$$\text{s.t.} \quad \sum_{c \in \mathcal{C}} f_{ce} \left(\sum_{a \in A^+_{vc}} y_{ac} - \sum_{a \in A^-_{vc}} y_{ac} \right) \leq r_{ve}, \quad v \in V,\ e \in \mathcal{C}_v \cap \mathcal{E}, \tag{3.2}$$

$$\sum_{c \in \mathcal{C}} f_{ce} \left(\sum_{a \in A^-_{vc}} y_{ac} - \sum_{a \in A^+_{vc}} y_{ac} \right) \leq r_{ve}, \quad v \in V,\ e \in \mathcal{C}_v \cap \mathcal{E}, \tag{3.3}$$

$$\sum_{c \in \mathcal{C}_a} y_{ac} = 1, \quad a \in A, \tag{3.4}$$

$$y_{ac} \in \{0, 1\}, \quad a \in A,\ c \in \mathcal{C}_a. \tag{3.5}$$

Constraints (3.2) and (3.3) ensure that r_{ve} is set to the net surplus or deficit of equipment type e at vertex v, while Constraint (3.4) guarantees that exactly one equipment type is assigned to each arc (i.e., the assigned equipment type remains the same or is replaced by exactly one other equipment type). Note that the minimum imbalance induced by a plan depends on both the loads and the initial assignment, $\mathscr{A}_0$, since $\mathcal{C}_a$ depends on $\mathscr{A}_0$.

Phase 2: Minimizing the number of changes required to achieve the minimum imbalance

In Phase 2, we minimize the number of changes Ω and adding Constraint (3.6) ensures that the obtained equipment assignment is an optimal assignment:

$$\begin{aligned} \Omega = \ & \min \sum_{a \in A} (1 - y_{a\mathscr{A}_0(a)}) \\ & \text{s.t.} \\ & \sum_{v \in V} \sum_{e \in \mathcal{C}_v \cap \mathcal{E}} r_{ve} \leq I^*, \qquad (3.6) \\ & (3.2), (3.3), (3.4), (3.5). \end{aligned}$$

Note that $\Omega < m$, and, thus, the two optimization models can be combined into a single optimization model as follows:

$$\begin{aligned} \min \quad & \sum_{v \in V} \sum_{e \in \mathcal{C}_v \cap \mathcal{E}} m r_{ve} + \sum_{a \in A} (1 - y_{a\mathscr{A}_0(a)}) \\ \text{s.t.} \quad & (3.2), (3.3), (3.4), (3.5). \end{aligned}$$

However, for real-life instances, m can be as large as hundreds of thousands, which makes it more difficult to solve than the two-stage hierarchical optimization model.

3.2.3 Stage 2: Restoring the remaining imbalance with empty repositioning

After minimizing the imbalance in Stage 1 by exhausting all the feasible equipment substitutions, it is possible that the final imbalance is still nonzero (i.e., $I^* > 0$). This implies that in order to further decrease the imbalance, we need other levers such as redistributing equipment in the network by introducing additional empty trailer movements from facilities with an outstanding surplus to facilities with an outstanding deficit. This is the objective of Stage 2 where we solve a minimum cost network flow problem for each equipment type to reduce the remaining imbalance I^* to zero

with the least additional repositioning cost (measured in terms of additional miles driven). In the following, we give the formulation of Stage 2.

For $v \in V$, let V_v^+ and V_v^- be the set of facilities that receive loads from facility v and the set of facilities that send loads to facility v, respectively. We define the following repositioning variable. For $v \in V$, $d \in V_v^+$ and $e \in \mathcal{E}$, let t_{vde} be the number of new empty trailer movements with equipment type e that we send from facility v to facility d. Let D_{vd} represent the distance between facility v and facility d. Stage 2 can be formulated as follows.

$$\min \sum_{v \in V} \sum_{d \in V_v^+} \sum_{e \in \mathcal{E}} D_{vd} t_{vde} \tag{3.7}$$

$$\text{s.t.} \left(\sum_{\substack{a \in \delta_v^+ \\ c = \mathscr{A}^*(a)}} f_{ce} + \sum_{d \in V_v^+} t_{vde} \right) - \left(\sum_{\substack{a \in \delta_v^- \\ c = \mathscr{A}^*(a)}} f_{ce} + \sum_{d \in V_v^-} t_{dvk} \right) = 0, \quad v \in V,\ e \in \mathcal{C}_v \cap \mathcal{E}, \tag{3.8}$$

$$t_{vde} \in \mathbb{Z}_{\geq 0}, \quad v \in V, d \in V_v^+, e \in \mathcal{E} \tag{3.9}$$

$$\tag{3.10}$$

The objective (3.7) minimizes the total distance travelled by new empty trailer movements (in miles). Constraints (3.8) represent the flow balance constraints for each facility-equipment pair. This model can be decomposed into $|\mathcal{E}|$ independent minimum cost flow problems.

3.3 Integrated Approach

In the staged approach, substitution and empty repositioning decisions are decoupled. In Stage 1, we minimize network imbalance using only equipment substitutions, and in Stage 2 we eliminate

the remaining imbalance by introducing new empty loads. While this staged approach can give very good results and can be implemented efficiently, it is a heuristic and does not guarantee that zero-imbalance is achieved with the minimum possible new empty load miles. To find an optimal repositioning plan, we have to formulate an optimization model that integrates both equipment substitution and empty repositioning decisions. We formulate a hierarchical integrated optimization model where in the first phase we minimize the total repositioning cost (i.e., the new empty load miles) to reach zero-imbalance, and in the second phase, we minimize the number of substitutions required to do so.

3.3.1 Formulation

We present the new formulations of Phases 1 and 2 of the integrated model. We use the same notation from Section 3.2.

Phase 1: Minimizing empty repositioning cost

$$\Delta_1 = \min \sum_{v \in V} \sum_{d \in V_v^+} \sum_{e \in \mathcal{E}} D_{vd} t_{vde} \tag{3.11}$$

$$\text{s.t.} \sum_{c \in \mathcal{C}} f_{ce} \left(\sum_{a \in A_{vc}^+} y_{ac} - \sum_{a \in A_{vc}^-} y_{ac} \right) + \left(\sum_{d \in V_v^+} t_{vde} - \sum_{d \in V_v^-} t_{dve} \right) = 0 \quad v \in V,\ e \in \mathcal{E}, \tag{3.12}$$

$$\sum_{c \in \mathcal{C}_a} y_{ac} = 1, \quad a \in A, \tag{3.13}$$

$$y_{ac} \in \{0,1\}, \quad a \in A,\ c \in \mathcal{C}_a. \tag{3.14}$$

$$t_{ide} \in \mathbb{Z}_{\geq 0}, \quad i, d \in V, e \in \mathcal{E}. \tag{3.15}$$

Δ_1 in (3.11) represents the objective function of Phase 1, which is to minimize the total empty repositioning miles. Constraints (3.12) represent the flow balance constraints for each facility-equipment type pair. These include also the empty repositioning variables t_{ide}. Constraints (3.13) ensure that the equipment type of load j can be substituted to only one other equipment type.

Phase 2: Minimizing the number of equipment substitutions

$$\begin{aligned} &\min \sum_{a \in A} (1 - y_{a\mathscr{A}_0(a)}) \\ &\text{s.t.} \sum_{v \in V} \sum_{d \in V_v^+} \sum_{e \in \mathcal{E}} D_{vd} t_{vde} \leq \Delta_1, \qquad (3.16) \\ &\qquad (3.12), (3.13), (3.14), (3.15). \end{aligned}$$

In Phase 2, we minimize the number of equipment substitutions required to reach zero-imbalance with the target the total number of miles Δ_1 reached in Phase 1. Constraint (3.16) ensures the target minimum total number of miles is satisfied.

Both Phase 1 and Phase 2 of the integrated model can be shown to be NP-hard; see Appendix D.

3.3.2 Staged vs Integrated Approach

We use an example with 6 facilities and 2 equipment types (orange and blue) as shown in Figure 3.1. The initial imbalance is 6 (4 for orange equipment and 2 for blue equipment). We assume full interchangeability between the two equipment types.

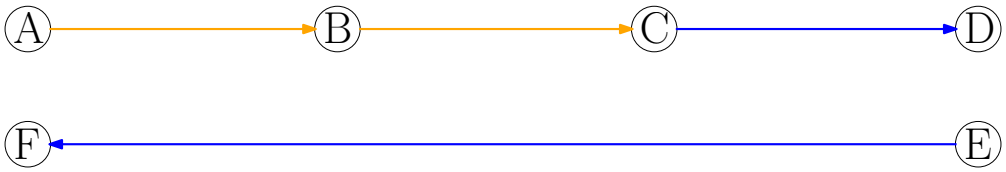
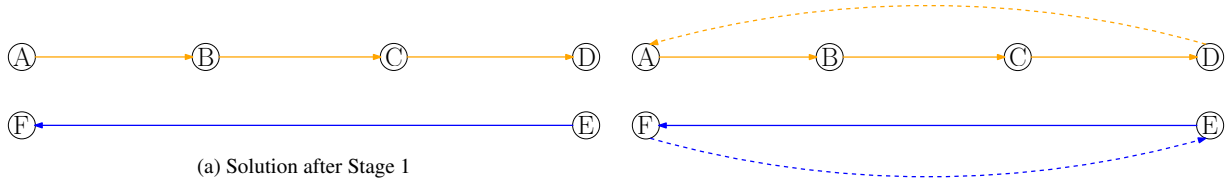


Figure 3.1: Example 1 of an imbalanced load plan with two equipment types

If we adopt the staged approach, the only optimal solution is given in Figure 3.2. The dashed arrows are the new empties added. The minimum imbalance after Stage 1 is 4 (2 for blue and 2 for orange equipment type) and it required one equipment substitution only (blue equipment on load from to C to D is substituted with orange equipment). Stage 2 reduces the imbalance to zero by adding two long empty loads, one blue from F to E and one orange from D to A.



(a) Solution after Stage 1

(b) Solution after Stage 2

Figure 3.2: Solution produced with the staged approach

If we use the integrated model, we get one optimal solution in Figure 3.3. Imbalance is reduced to zero using 2 substitutions and 2 short empty loads. Notice that we can construct examples where the solution achieved by the staged approach required a high repositioning cost as compared to the integrated model. This shows that the staged approach can perform poorly compared to an integrated model.

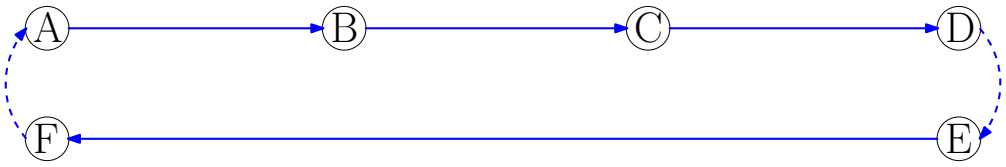


Figure 3.3: Example of an optimal solution of the integrated model.

There are also situations where addressing imbalance by empty repositioning only, i.e., solving only Stage 2 of the staged approach, can yield a lower repositioning cost than the staged approach.

Consider a situation with 5 facilities as shown in Figure 3.4. The initial imbalance is 6 (2 for orange equipment and 4 for blue equipment). We again assume full interchangeability between the two equipment types.

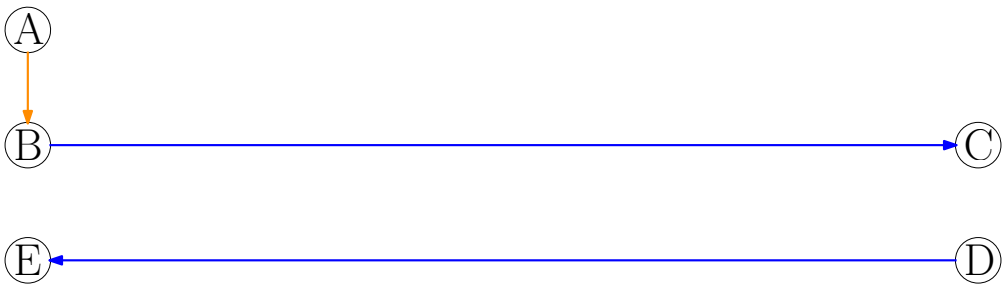
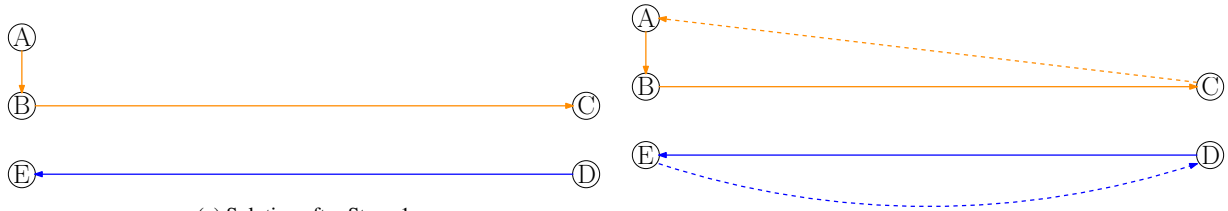


Figure 3.4: Example 2 of an imbalanced load plan with two equipment types

If we adopt the staged approach, one optimal solution is given in Figure 3.5. The minimum imbalance after Stage 1 is 4 (2 for blue and 2 for orange equipment type) and it requires one equipment substitution (blue equipment on load from to B to C is substituted with orange equipment). Stage 2 reduces the imbalance to zero by adding two long empty loads, one blue from E to D and one orange from C to A.



(a) Solution after Stage 1

(b) Solution after Stage 2

Figure 3.5: Solution produced with the staged approach for Example 2

If we address imbalance by empty repositioning only (solving only Stage 2), we get the optimal solution in Figure 3.6. Imbalance is reduced to zero using 3 short empty loads (one orange from B to A and two blue from C to D and from E to B.

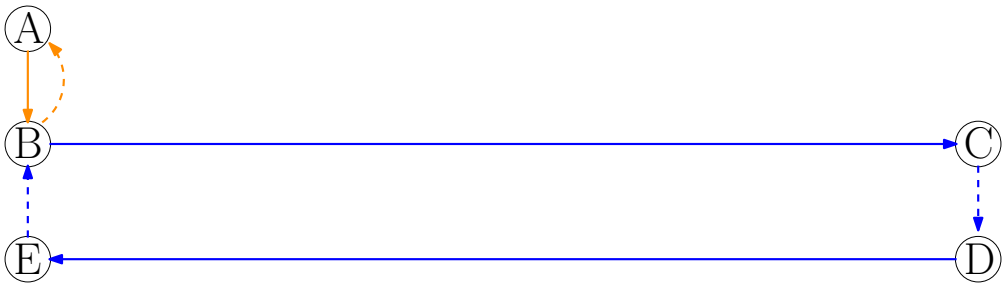


Figure 3.6: Example of an optimal solution with empty repositioning alone.

This shows that the staged approach is not guaranteed to produce high-quality solutions (even when each stage is solved to optimality).

3.4 Computational study

Our computational study uses instances derived from weekly load plans and weekly driver schedules for a package express network.

3.4.1 Equipment and substitution matrices

There are three categories of equipment: *Shorts* (**S**), *Longs* (**L**), and *Extra Longs* (**XL**). Category **S** contains three equipment types: W, WW, and SC, category **L** contains nine equipment types: Z, ZZ, S, Y, YY, O, OO, TMF, and TMB, and category **XL** contains two equipment types: ZZZ and LC. To support strategic and tactical analysis, it is beneficial to be able to accommodate different sets of substitution rules. This is accomplished by the introduction of an equipment substitution matrix (ESM). An ESM is 0-1 matrix, where a 1 in the i-th row and j-th column means that the equipment type corresponding to the i-th row can be substituted with the equipment type corresponding to the j-th column, and a 0 means that this substitution is not allowed. To determine the set of allowable equipment types $\mathcal{C}_a$ for a load a, we consider the equipment substitution matrix, whether the facilities at the origin and destination of the load have equipment type restrictions, and the size of the load (a smaller equipment type is allowed only if the load fits). The most restrictive

equipment substitution matrix, ESM1, is shown in Table 3.1. It does not consider composite equipment types and it does not allow substituting a short equipment type with a larger equipment type (i.e., and equipment type in **L** or **XL**). The latter requirement tries to avoid the use of lightly utilized equipment since the size of equipment in category **S** is much smaller that the size of equipment in categories **L** and **XL**. Note that equipment types TMF and TMB are only allowed to be swapped to each other.

Table 3.1: ESM1

	W	WW	SC	Z	ZZ	S	Y	YY	O	OO	TMF	TMB	ZZZ	LC
W	1	1	1	0	0	0	0	0	0	0	0	0	0	0
WW	1	1	1	0	0	0	0	0	0	0	0	0	0	0
SC	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Z	1	1	1	1	1	1	1	1	1	1	0	0	1	1
ZZ	1	1	1	1	1	1	1	1	1	1	0	0	1	1
S	1	1	1	1	1	1	1	1	1	1	0	0	1	1
Y	1	1	1	1	1	1	1	1	1	1	0	0	1	1
YY	1	1	1	1	1	1	1	1	1	1	0	0	1	1
O	1	1	1	1	1	1	1	1	1	1	0	0	1	1
OO	1	1	1	1	1	1	1	1	1	1	0	0	1	1
TMF	0	0	0	0	0	0	0	0	0	0	1	1	0	0
TMB	0	0	0	0	0	0	0	0	0	0	1	1	0	0
ZZZ	1	1	1	1	1	1	1	1	1	1	0	0	1	1
LC	1	1	1	1	1	1	1	1	1	1	0	0	1	1

Equipment substitution matrix ESM2 introduces composite equipment type 2WW, which consists of two shorts, specifically two pieces of equipment type WW, with capacity close to the capacity of the equipment types in category **XL**. As shown in Table 3.2, all equipment types, except TMF and TMB, can be substituted with equipment type 2WW, which gives much more flexibility, but, as a result, also makes the optimization models more difficult to solve. As none of the loads in the system initially have a composite equipment type, there is no need to include a row for equipment type 2WW in the equipment substitution matrix.

Equipment substitution matrix ESM3 introduces bobtails (BT). When a load is labeled as a bobtail, it means that a tractor moves without pulling any trailer. Assigning an equipment type to a bobtail, i.e., having the tractor pull one or more (empty) trailers, can be an effective way to reduce

Table 3.2: ESM2

	W	WW	SC	Z	ZZ	S	Y	YY	O	OO	TMF	TMB	ZZZ	LC	2WW
W	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
WW	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
SC	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
Z	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
ZZ	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
S	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
Y	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
YY	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
O	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
OO	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
TMF	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
TMB	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
ZZZ	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
LC	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1

the imbalance. Furthermore, we allow short equipment types to be substituted with large or extra large equipment types.

Table 3.3: ESM3

	W	WW	SC	Z	ZZ	S	Y	YY	O	OO	TMF	TMB	ZZZ	LC	2WW	BT
W	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
WW	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
SC	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
Z	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
ZZ	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
S	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
Y	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
YY	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
O	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
OO	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
TMF	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
TMB	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
ZZZ	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
LC	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
BT	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

3.4.2 Instances

We use ten instances in our computational experiments derived from weekly load plans and driver schedules. Detailed information about the instances can be found in Table 4.1. Each weekly load

plan contains all the loads that are scheduled to be dispatched during the week. The loads are of three types: (a) loaded, in which case an equipment type is assigned and a volume is specified (as a percentage of trailer capacity), (b) empty, in which case an equipment type is assigned, but there is no volume specified, and (c) bobtail, in which case no equipment type is specified. The instances have about 3,350 facilities and about 300 thousand loads, with about 40% of these being loaded, 30% being empty, and 30% being bobtails. The initial imbalance in the instances is around 4,000. The load plans have been modified slightly, to ensure that balance can be restored. (In Appendix C, we present necessary and sufficient conditions to guarantee that balance can be restored.) The fleet is composed of 15 different types of trailers and containers. All the models are coded in C++.

Table 3.4: Information on the instances used in the computational experiments.

Instance	# Loads	# Facilities	# Total Miles	Initial Imbalance	Empty Loads (%)	Bobtails (%)
I1	302,344	3,254	39,768,799	4,098	29.40	28.14
I2	301,509	3,982	39,680,301	3,982	29.33	28.05
I3	301,270	3,270	39,687,329	4,006	29.19	28.08
I4	300,963	3,273	39,658,470	4,106	29.09	28.10
I5	300,298	3,285	39,653,034	4,126	29.03	27.95
I6	300,013	3,275	39,708,418	3,948	28.93	27.85
I7	299,519	3,281	39,697,927	4,078	28.80	27.79
I8	299,519	3,280	39,766,442	3,870	28.83	27.63
I9	299,107	3,291	39,801,865	3,786	28.89	27.38
I10	299,415	3,285	39,817,469	3,566	28.89	27.38

Gurobi 8.1 with default settings is used for solving the mixed integer programs. All experiments were run in a 20-core machine with Intel(R) Xeon(R) 2.30GHz processors and 256GB of RAM. The optimality tolerance is set to 0.005 for Phase 1 and 0.05 for Phase 2 in the first stage of the staged approach, and 0.005 for the two phases of the integrated approach. No time limit was enforced.

3.4.3 Two simple decomposition heuristics

Substitution decomposition

When the composite equipment type 2WW is not allowed (ESM1), the coefficient matrix of the optimization problems presented in Section 3.2.2 and 3.2.2 will be a (0,1)-matrix, and the optimization problems can be solved relatively easily in practice (even though the problem is **NP-hard**). However, when the composite equipment type 2WW is allowed, the coefficient matrices will no longer be (0,1)-matrices, as the composite equipment type consists of two basic equipment types. As a consequence, solving the optimization problems is much more difficult and requires much more computing time. Therefore, we have developed a simple, but computationally effective, two-phase decomposition heuristic that we call SUB-HEUR. In the first phase, we only allow equipment to be substituted with short or composite equipment types (**S** and **C**), which makes the optimization problem more tractable. In the second phase, the substitutions identified in the first phase are fixed and we seek to further reduce the imbalance by equipment substitutions among the **S**, **L**, and **XL** categories. Computational experiments show that SUB-HEUR heuristic produces high-quality solutions in a short amount of time.

Spatial decomposition

One natural factor that makes the substitution based models hard to solve is the size of the service networks (more than 3,000 nodes in the instances we are solving). This yields very large scale mixed integer models that commercial solvers can't solve efficiently to optimality especially for substitutions matrices that involve substitutions to composite configurations. More particularly, solving just the LP relaxation of the IP models is computationally hard, let alone the discrete models. We explore here heuristics that decompose the problem into relatively small subproblems that a commercial solver can solve in reasonable amount of time.

We exploit the already existing partition of the small package network intro regional divisions

referred to as *districts*. The idea is to solve the optimization model for each district independently (either sequentially or in parallel) while keeping the initial equipment types in load chains between districts unchanged in a first stage. In the second stage, we minimize equipment imbalance using only load chains between districts. We refer to load chains inside a district as *intra-district* arcs and the load chains between districts as *inter-district* arcs. We will explore two variants of the heuristic depending on the order in which we solve intra-district and inter-district models.

Note that one could use a different clustering approach to partition the network into sub-problems similar in size such that the inter-cluster arcs are minimized. This could yield better solutions as the inter-cluster model will be small.

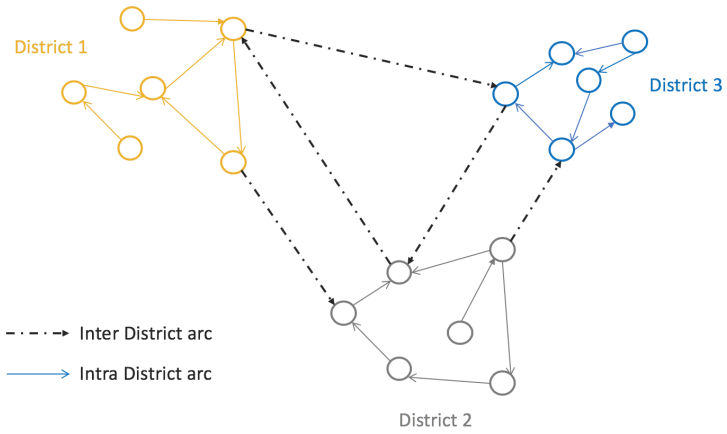


Figure 3.7: Representation of districts

To solve a district independently, we consider the sub-network of the facilities that constitute the district of interest. We add two artificial facilities, a source and a sink. All the inter-district arcs that are inbound to the district of interest are assumed to depart from the source node, and all the inter-district arcs that depart from the district of interest are assumed to be bound to the sink node. In the optimization models, no substitution of equipment is allowed on the inter-district arcs both departing or arriving at the district of interest. Also, imbalance is not considered at both the source and the sink nodes. Finally, the optimization models used here are similar to the original models for the whole network.

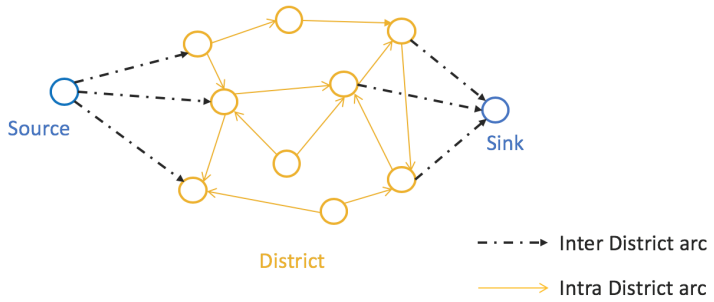


Figure 3.8: example of a district

In the first variant INTRA-FIRST HEURISTIC, we first minimize equipment imbalance within each district, then we fix the intra-district arcs and minimize the equipment imbalance between districts. We can iterate this process multiple times. Algorithm 8 gives the pseudo-code.

Algorithm 8: INTRA-FIRST HEURISTIC(N_{iter})

Districts $\leftarrow$ partition the network of facilities into districts

$k \leftarrow 0$

while $k < N_{iter}$ **do**

  $k \leftarrow k + 1$

  for ***each*** *district* ***d*** *in* ***Districts*** **do**

    Fix equipment type in inbound/outbound inter-district arcs to district **d**

    Solve the hierarchical model to minimize imbalance within district **d**

    Apply the suggested substitutions to intra-district arcs within district **d**

  Fix equipment type in intra-district arcs

  Solve the hierarchical model to minimize imbalance between **Districts**

  Apply the suggested substitutions to inter-district arcs

In the second variant INTER-FIRST HEURISTIC, we change the order and we first minimize equipment imbalance between all the districts, then we fix the inter-district arcs and minimize the equipment imbalance within each district separately. We can iterate this process multiple times. Algorithm 9 gives the pseudo-code.

Algorithm 9: INTER-FIRST HEURISTIC(N_{iter})

Districts $\leftarrow$ partition the network of facilities into districts

$k \leftarrow 0$

while $k < N_{iter}$ **do**

  $k \leftarrow k + 1$

  Fix equipment type in intra-district arcs

  Solve the hierarchical model to minimize imbalance between **Districts**

  Apply the suggested substitutions to inter-district arcs

  for ***each*** *district* ***d*** *in* ***Districts*** **do**

    Fix equipment type in inbound/outbound inter-district arcs to district **d**

    Solve the hierarchical model to minimize imbalance within district **d**

    Apply the suggested substitutions to intra-district arcs within district **d**

3.4.4 Analysis

In the first set of the experiments, we focus on Stage 1 of the staged approach, i.e., minimizing imbalance with a minimum number of substitutions. Furthermore, we evaluate the performance of the substitution decomposition heuristic SUB-HEUR as this heuristic performs better than the spatial decomposition based heuristics in terms of solution quality and run-time (we report the results of the latter ones in Appendix B). The difference in performance is due to the fact that we solve the sub-problems in INTER-FIRST and INTRA-FIRST sequentially (although they can be solved in parallel as they are independent) and that we use the partition of the network, in terms of geographical districts, provided by the company. A smart clustering and the use of parallel computing could improve the performance of the spatial decomposition based heuristic. In the second set of experiments, we focus on Stage 2 of the staged approach and assess the benefits of resorting to empty repositioning after exhausting equipment substitutions to restore full balance. In the third set of experiments, we assess the value of the integrated model and compare its performance to the

staged approach.

Staged Approach: Stage 1

Let I_0 and $\hat{I}$ be the initial imbalance and minimum imbalance, respectively, and let the imbalance reduction be $\Delta I = (I_0 - \hat{I})/I_0$. Let N_s be the number of substitutions in the Phase 1 solution, let $\hat{N}_s$ be the minimum number of substitutions required to reach minimum imbalance, and let the substitution reduction be $\Delta N_s := (N_s - \hat{N}_s)/N_s$.

Table 3.5 shows the optimization results using ESM1. We observe that, on average, the imbal-

Table 3.5: Optimization results using equipment substitution matrix ESM1.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	3,600	12.15	18	50,441	275	99.45	139
I2	3,982	3,474	12.76	16	50,858	268	99.47	176
I3	4,006	3,476	13.23	19	50,032	295	99.41	255
I4	4,106	3,542	13.74	20	50,094	289	99.42	783
I5	4,126	3,564	13.62	21	49,554	316	99.36	110
I6	3,948	3,426	13.22	16	49,857	311	99.38	221
I7	4,078	3,600	11.72	17	48,765	288	99.41	154
I8	3,870	3,498	9.61	17	49,346	222	99.55	60
I9	3,786	3,450	8.87	16	50,809	192	99.62	81
I10	3,566	3,344	6.23	15	49,298	165	99.67	202

ance can be reduced by about 11% and that this requires, on average, fewer than 300 substitutions. We also observe that the Phase 2 optimization takes, on average, more than 10 times as long as the Phase 1 optimization.

Tables 3.6 and 3.7 show the optimization and heuristic results using ESM2, respectively. ESM2 allows, on top of ESM1, the possibility to substitute a load with composite equipment type 2WW, which provides more flexibility.

We observe that, on average, the imbalance can be reduced by about 50% and that this requires, on average, fewer than 3000 substitutions. We also observe that the decomposition heuristic per-

Table 3.6: Optimization results using equipment substitution matrix ESM2.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	2,010	50.95	1,391	111,817	2,851	97.45	8,623
I2	3,982	1,894	52.44	4,257	120,030	2,817	97.65	50,338
I3	4,006	1,835	54.19	1,656	114,860	2,939	97.44	8,937
I4	4,106	1,913	53.41	822	114,541	2,928	97.44	43,061
I5	4,126	1,968	52.30	1,669	115,582	2,885	97.50	18,931
I6	3,948	1,916	51.47	2,410	114,831	2,735	97.62	11,581
I7	4,078	2,147	47.35	2,161	115,228	2,811	97.56	6,413
I8	3,870	2,075	46.38	2,017	116,725	2,933	97.49	18,702
I9	3,786	2,036	46.22	2,158	112,962	2,813	97.51	9,489
I10	3,566	1,882	47.22	3,817	112,454	2,982	97.35	15,713

Table 3.7: Heuristic results using equipment substitution matrix ESM2.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	2,092	48.95	42	54,763	2,791	94.90	152
I2	3,982	1,968	50.58	44	55,247	2,845	94.85	150
I3	4,006	1,941	51.55	47	55,982	2,846	94.92	277
I4	4,106	1,971	52.00	44	55,597	2,937	94.72	229
I5	4,126	2,034	50.70	43	55,478	2,887	94.80	191
I6	3,948	1,977	49.92	44	56,446	2,813	95.02	361
I7	4,078	2,217	45.64	43	55,514	2,758	95.03	242
I8	3,870	2,150	44.44	47	55,770	2,866	94.86	292
I9	3,786	2,080	45.06	47	56,572	2,863	94.94	265
I10	3,566	1,948	45.37	51	56,181	2,859	94.91	187

forms well, achieving, on average, an imbalance reduction of about 48% and also requiring, on average, fewer than 3000 substitutions to achieve this. However, we see a dramatic reduction in computing time, i.e., on average by more than 98%.

Tables 3.8 and 3.9 show the optimization and heuristic results, respectively, using equipment substitution matrix ESM3. ESM3 allows, on top of ESM2, the substitution of a bobtail movement to any equipment type.

Table 3.8: Optimization results using equipment substitution matrix ESM3.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	1,306	68.13	32,710	196,070	3,394	98.27	97,721
I2	3,982	1,228	69.16	38,124	161,337	3,519	97.82	159,458
I3	4,006	1,225	69.42	29,418	170,337	3,422	97.99	49,144
I4	4,106	1,213	70.46	16,373	170,302	3,485	97.95	76,702
I5	4,126	1,262	69.41	18,206	176,428	3,457	98.04	100,794
I6	3,948	1,176	70.21	21,099	166,709	3,581	97.85	39,000
I7	4,078	1,439	64.71	23,640	163,384	3,456	97.88	144,742
I8	3,870	1,387	64.16	18,394	170,292	3,500	97.94	162,128
I9	3,786	1,321	65.11	40,144	163,418	3,768	97.69	85,082
I10	3,566	1,215	65.93	36,830	188,755	3,877	97.95	187,588

Table 3.9: Heuristic results using equipment substitution matrix ESM3.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	1,552	62.13	590	58,964	3,423	94.19	1,101
I2	3,982	1,474	62.98	567	58,187	3,342	94.26	908
I3	4006	1451	63.78	672	55323	3179	94.25	884
I4	4,106	1,477	64.03	639	57,882	3,303	94.29	971
I5	4,126	1,514	63.31	1,696	191,098	14,334	92.50	168
I6	3,948	1,437	63.60	717	58,261	3,367	94.22	960
I7	4,078	1,673	58.97	1,336	191,357	13,893	92.74	131
I8	3,870	1,645	57.49	914	57,730	3,119	94.60	1,609
I9	3,786	1,575	58.40	831	59,217	3,308	94.41	727
I10	3,566	1,418	60.24	1,595	189,423	14,467	92.36	130

We observe that, on average, the imbalance can be reduced by about 67% and that this requires, on average, about 3,500 substitutions. We also observe that the decomposition heuristic continues to perform well in terms of imbalance reduction, achieving, on average, a reduction of about 62%. However, its behavior varies in terms of number of substitutions required. In three of the ten instances, the number of substitutions required is around 14,000. On the other hand, optimization starts to become computationally prohibitive with some instances requiring more than 60 hours of computing (Phase 1 plus Phase 2). Most instances require less than 30 minutes of computing time using the decomposition heuristic.

For ease of comparison, we present a few critical statistics for the different equipment substitution matrices in Table 3.10. These statistics clearly show the benefits derived from allowing

Table 3.10: A few critical statistics for substitution matrices ESM1, ESM2 and ESM3.

Instance	ESM1		ESM2				ESM3			
	Exact		Exact		Heuristic		Exact		Heuristic	
	$\Delta I(\%)$	$\hat{N}_s$	$\Delta I(\%)$	$\hat{N}_s$	$\Delta I(\%)$	$\hat{N}_s$	$\Delta I(\%)$	$\hat{N}_s$	$\Delta I(\%)$	$\hat{N}_s$
I1	12.15	275	50.95	2,851	48.95	2,791	68.13	3,394	62.13	3,423
I2	12.76	268	52.44	2,817	50.58	2,845	69.16	3,519	62.98	3,342
I3	13.23	295	54.19	2,939	51.55	2,846	69.42	3,422	63.78	3,179
I4	13.74	289	53.41	2,928	52.00	2,937	70.46	3,485	64.03	3,303
I5	13.62	316	52.30	2,885	50.70	2,887	69.41	3,457	63.31	14,334
I6	13.22	311	51.47	2,735	49.92	2,813	70.21	3,581	63.60	3,367
I7	11.72	288	47.35	2,811	45.64	2,758	64.71	3,456	58.97	13,893
I8	9.61	222	46.38	2,933	44.44	2,866	64.16	3,500	57.49	3,119
I9	8.87	192	46.22	2,813	45.06	2,863	65.11	3,768	58.40	3,308
I10	6.23	165	47.22	2,982	45.37	2,859	65.93	3,877	60.24	14,467

more flexible substitution rules and that the performance of the decomposition heuristic is good, but deteriorates slightly when the flexibility increases.

We next explore in detail for a single instance the relationship between the imbalance reduction and the number of substitutions required to achieve that reduction. More specifically, Figure 3.9 shows this relationship for Instance 1 (using ESM3). We see that the closer we get to the maximum

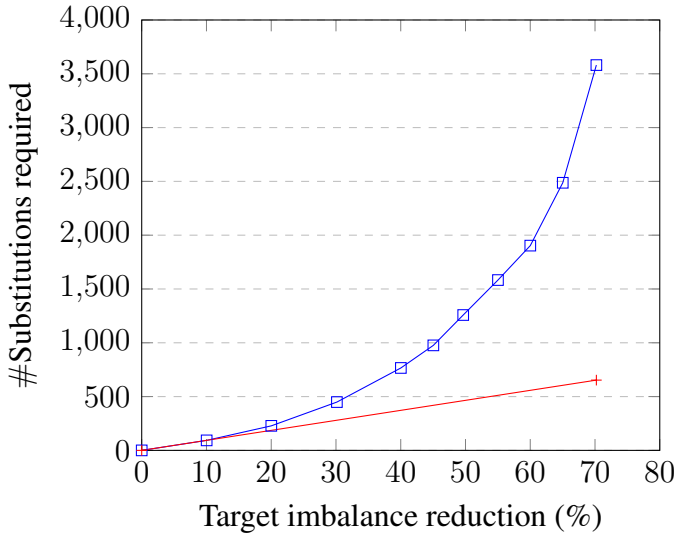


Figure 3.9: Relationship between the imbalance reduction and the number of substitutions required for Instance 1

possible imbalance reduction, the larger the number of substitutions required. For instance, to reduce the initial imbalance by 50%, about 1,250 substitutions are required. However, to reduce the imbalance from 50% to 70%, about 2,300 additional substitutions are required. The slope at 0 (in red) is equivalent to about 0.25 substitutions per unit of imbalance resolved. This was expected as the model tries to maximize the substitutions that have the highest impact which is resolving 4 units of imbalance with a one-to-one substitution.

Staged Approach: restoring balance in Stage 2

Next, we assess the benefits of complementing empty repositioning with equipment substitution. To do so, we compare two scenarios: (a) restoring balance of a given load plan by adding empty loads, and (b) minimizing the imbalance of a given load plan by substituting equipment and then adding empty loads. In both scenarios, when adding empty loads, we minimize the number of empty load miles added (this is a minimum cost flow problem). When minimizing the imbalance of a given load plan by substituting equipment, we use equipment substitution matrix ESM3. In Table 3.11, we report for both scenarios the number of empty loads added and the number of empty

load miles added as well as the reduction in empty load miles when equipment substitutions are performed before empty loads are added.

Table 3.11: Restoring balance with and without equipment substitutions.

Instance	Empty repositioning		Equipment substitutions + Empty repositioning		Miles reduction (%)
	# Empties	# Miles (%)[a]	# Empties	# Miles (%)[a]	
I1	14,082	52,523 (0.13)	6,930	30,559 (0.08)	41.82
I2	14,012	51,460 (0.13)	6,799	32,185 (0.08)	37.46
I3	13,548	48,996 (0.12)	6,237	29,675 (0.07)	39.43
I4	14,154	50,382 (0.13)	6,552	29,713 (0.07)	41.02
I5	14,189	48,715 (0.12)	7,073	32,617 (0.08)	33.05
I6	13,392	44,613 (0.11)	6,493	29,125 (0.07)	34.72
I7	14,730	52,757 (0.13)	7,647	34,013 (0.09)	35.53
I8	14,477	62,710 (0.16)	7,905	45,398 (0.11)	27.61
I9	14,172	50,478 (0.13)	7,593	33,509 (0.08)	33.62
I10	12,956	45,024 (0.11)	6,960	28,603 (0.07)	36.47

[a] Percentage of Total Miles

We observe that the number of empty loads that need to be added to restore balance reduces significantly when we first reduce the imbalance by equipment substitutions, and, more importantly, that the number of empty load miles added reduces significantly, on average by about 36%.

Integrated Model

The staged approach mimics what happens in practice as companies' primary focus is minimizing equipment imbalance and only afterwards and not necessarily always reducing any remaining imbalance by empty repositioning. Here, we explore the value of integrating equipment substitutions and empty repositioning into a single optimization model with the objective to attain zero imbalance with the least cost (in terms of required empty repositioning).

To assess the value of the integrated model, we compare the performance of the following schemes:

- EMPTY-REPOS: We minimize imbalance by means of empty repositioning only without resorting to equipment substitution decisions. This boils down to running Stage 2 only of the staged approach,
- STAGED-EXACT: We run the two stages of the staged approach. Stage 1 models are solved exactly,
- STAGED-HEUR: We run the two stages of the staged approach. Stage 1 models are solved with the substitution decomposition based heuristic SUB-HEUR,
- INTEGRATED-EXACT: We minimize imbalance using the integrated model. Phase 1 and 2 are solved exactly,
- INTEGRATED-HEUR: We minimize imbalance using the integrated model. Phase 1 and 2 are solved with SUB-HEUR heuristic.

For this experiment, we used a different set of instances than the ones used in the first and second experiment. Table 3.12 summarizes some key statistics on the new set of instances. The fleet in these instances is composed of 19 different types of trailers and containers.

Table 3.12: Information on the instances used in the Integrated Model experiment.

Instance	# Loads	# Balance Facilities	# Total Miles	Initial Imbalance	Empty Loads (%)	Bobtails (%)
I11	588,847	7,372	58,245,354	20,602	31.09	23.56
I12	573,969	7,366	57,055,192	18,646	31.12	23.19
I13	500,129	7,211	53,830,514	18,014	30.97	20.93
I14	267,663	1,426	39,074,866	24,450	34.76	14.14
I15	263,394	1,426	38,159,237	24,064	34.20	14.60
I16	231,090	1,417	35,751,361	21,924	33.13	12.65

Table 3.13 summarizes the results of the comparison of the different schemes. We report the total number of miles required to reach the minimum equipment imbalance (**#Miles**), the imbalance

reduction ($\Delta I_1(\%)$) due to equipment substitutions only. This is the imbalance achieved by Stage 1 for the staged approach. For the integrated model, this represents the imbalance we see if we ignore the empty repositioning component in the solution. We also report the minimum number of substitutions used (N_s^*), and the run time (**Time**) in seconds.

The staged approach STAGED-EXACT reduces the repositioning cost by about 35% as compared to using empty repositioning alone (EMPTY-REPOS). This difference is more pronounced for Instances 14-16 where the reduction is about 58%. This can be explained by the performance of Stage 1 of the staged approach that accounts for about 87% reduction in imbalance for Instances 14-16 compared to only 24% for Instances 11-13. We see the value of integrating both equipment substitutions and empty repositioning in INTEGRATED-EXACT which further reduces the empty repositioning cost by about 68% compared to STAGED-EXACT. In terms of the number of equipment substitutions required, as expected, the staged approach STAGED-EXACT requires about 51% fewer equipment substitutions than the integrated scheme INTEGRATED-EXACT. The difference is less pronounced for Instances 14-16. This is in line with the high imbalance reduction by equipment substitutions ($\Delta I_1(\%)$) that the scheme INTEGRATED-EXACT achieves for Instances 14-16. In terms of run time, INTEGRATED-EXACT requires about 4 times more computational effort than solving the staged approach STAGED-EXACT. For Instances 11-13, the integrated model created more imbalance with equipment substitutions in order to minimize the final repositioning cost that is the primary objective. For these instances, equipment substitutions increased imbalance by about 15% (this explains the negative values in Column $\Delta I_1(\%)$). This also explains the poor performance of the staged approach for these instances in terms of repositioning cost as the approach essentially minimizes imbalance with equipment substitutions in Stage 1 which may not be the optimal strategy.

We also see the value of using the substitution decomposition based heuristic SUB-HEUR in both the staged approach and the integrated model. For the staged approach, the heuristic yields comparable imbalance reduction in Stage 1 for all Instances except for Instance 16 (70.21% reduc-

Table 3.13: Results for the set of instances comparing different schemes of empty repositioning, staged approach, and integrated approach.

Instance	Scheme	#Miles (%)[a]	$\Delta I_1(\%)$	N_s^*	Time
I11	EMPTY-REPOS	520,847 (0.89)	-	-	1,183
	STAGED-EXACT	438,057 (0.75)	21.99	3,387	1,995
	STAGED-HEUR	429,181 (0.74)	21.56	7,184	2,723
	INTEGRATED-EXACT	139,468 (0.24)	-14.41	13,577	9,813
	INTEGRATED-HEUR	146,948 (0.25)	-18.35	19,535	5,875
I12	EMPTY-REPOS	498,250 (0.87)	-	-	1,121
	STAGED-EXACT	423,076 (0.74)	22.42	3,073	2,413
	STAGED-HEUR	421,829 (0.74)	21.74	3,951	2,743
	INTEGRATED-EXACT	133,517 (0.23)	-16.15	12,589	8,702
	INTEGRATED-HEUR	141,530 (0.25)	-21.23	19,112	6,097
I13	EMPTY-REPOS	482,884 (0.90)	-	-	1,138
	STAGED-EXACT	452,077 (0.84)	26.42	2,852	2,565
	STAGED-HEUR	471,570 (0.88)	25.61	5,285	1,588
	INTEGRATED-EXACT	139,398 (0.26)	-14.80	13,693	15,440
	INTEGRATED-HEUR	213,606 (0.40)	-27.21	18,484	9,907
I14	EMPTY-REPOS	1,224,408 (3.13)	-	-	371
	STAGED-EXACT	489,178 (1.25)	89.01	15,812	692
	STAGED-HEUR	500,651 (1.28)	88.44	18,459	370
	INTEGRATED-EXACT	135,276 (0.35)	87.28	20,286	3,324
	INTEGRATED-HEUR	145,401 (0.37)	84.33	27,663	1,799
I15	EMPTY-REPOS	1,212,630 (3.18)	-	-	484
	STAGED-EXACT	503,470 (1.32)	89.20	15,902	562
	STAGED-HEUR	474,275 (1.24)	88.62	18,608	475
	INTEGRATED-EXACT	132,733 (0.35)	87.67	20,289	2,039
	INTEGRATED-HEUR	143,677 (0.38)	84.67	27,085	1,956
I16	EMPTY-REPOS	1,170,753 (3.27)	-	-	480
	STAGED-EXACT	515,032 (1.44)	84.23	13,286	5,263
	STAGED-HEUR	758,936 (2.12)	70.21	12,733	813
	INTEGRATED-EXACT	225,166 (0.63)	77.70	18,993	9,155
	INTEGRATED-HEUR	432,625 (1.21)	58.99	22,846	2,586

[a] Percentage of Total Miles

tion as compared to 84.23% with the exact method). It also reduces the run-time by about 22% on average. For Instances 1 and 2 the heuristic required more computational time because of long run times in Stage 2; see Table 3.14 for details on run times. For the integrated approach, the heuristic increased the repositioning cost by about 29%, and required 37% more equipment substitutions compared to the exact method. The value of the heuristic is in the computational effort as it decreases the run time by about 38% compared to the exact method. Table 3.15 shows the comparison of run time for Phase 1 and Phase 2 separately. The heuristic decreased the run-time by about 42% for Phase 1 and 14% for Phase 2.

Table 3.14: Run-time (in seconds) of both Stages 1 and 2 of the schemes STAGED-EXACT and STAGED-HEUR. The run time excludes the time spent in the data processing

		Run time excluding data processing		
Instance	**Scheme**	**Stage 1/Phase 1**	**Stage 1/Phase 2**	**Stage 2**
I11	STAGED-EXACT	328	516	1,001
	STAGED-HEUR	228	88	2,134
I12	STAGED-EXACT	416	968	885
	STAGED-HEUR	168	560	1,766
I13	STAGED-EXACT	1,433	324	674
	STAGED-HEUR	637	129	699
I14	STAGED-EXACT	288	179	131
	STAGED-HEUR	44	98	141
I15	STAGED-EXACT	213	130	128
	STAGED-HEUR	52	144	151
I16	STAGED-EXACT	776	4,253	140
	STAGED-HEUR	70	271	276

In Table 3.16 and Table 3.17, we give some statistics for Instance I15 regarding the mileage workload change for the staged approach and the integrated model, respectively. The mileage workload for a given equipment type is computed as the total miles of the loads that use that equipment type. We report the change in the workload considering equipment substitution decisions only as well as considering both equipment substitution and empty repositioning decisions together with respect to the initial workload. For both the staged approach and the integrated model,

Table 3.15: Run-time (in seconds) of both Phases 1 and 2 of the schemes INTEGRATED-EXACT and INTEGRATED-HEUR. The run time excludes the time spent in the data processing.

Instance	Scheme	Run time excluding data processing	
		Phase 1	Phase 2
I11	INTEGRATED-EXACT	9,192	335
	INTEGRATED-HEUR	5,500	156
I12	INTEGRATED-EXACT	8,245	203
	INTEGRATED-HEUR	5,569	214
I13	INTEGRATED-EXACT	14,981	260
	INTEGRATED-HEUR	9,268	372
I14	INTEGRATED-EXACT	2,705	321
	INTEGRATED-HEUR	1,193	190
I15	INTEGRATED-EXACT	1,653	148
	INTEGRATED-HEUR	1,465	149
I16	INTEGRATED-EXACT	8,374	480
	INTEGRATED-HEUR	2,012	300

the workload of the equipment categories D40, MAR, Z53, and LC changes significantly, whereas the workload of REN, AIR, and PCA remains stable. The workload of CPU changes significantly in the integrated model but remains stable in the staged approach. MAR sees the same workload change in both models.

3.5 Final Remarks

Striving for equipment balance, i.e., seeking to have the same equipment at a facility at the end of the week as at the start of the week, ignores what happens during the week, and does not account for seasonal changes in package flows. A natural next research direction, therefore, is inventory-aware equipment management, in which time is modeled explicitly, e.g., days for planning periods of one or more weeks, and weeks for planning periods of one or more quarters. We explore this research direction in Chapter 4.

Table 3.16: Workload change with equipment substitution decision only, and with additional empty repositioning for instance I15 solved with the staged approach.

Equipment Category	Initial Workload	Post Substitution Workload	Change Percentage (%)	Post Empty Repositioning Workload	Change Percentage (%)
AIR	412,397	411,639	-0.18	413,386	0.24
CPU	154,846	144,635	-6.59	155,632	0.51
SPU	10,696	9,154	-14.42	11,076	3.56
D40	175	0	-100	0	-100
LC	290,981	239,689	-17.63	250,951	-13.76
MAR	5,876	3,649	-37.9	4,179	-28.88
MDV	0	0	0	0	0
PCA	1,111,500	1,096,950	-1.31	1,101,490	-0.90
PUP	27,657,600	28,529,000	3.15	28,842,400	4.28
Z53	6,142,480	4,705,920	-23.39	4,786,460	-22.08
REN	216	216	0	216	0
RR	80,035	71,797	-10.29	74,479.4	-6.94
Y	509,977	516,230	1.23	546,938	7.25
Z	800,648	750,746	-6.23	759,515	-5.14

Table 3.17: Workload change with substitutions decisions only, and with additional empty repositioning for instance I15 solved with the integrated model.

Equipment Type	Initial Workload	Post Substitution Workload	Change Percentage (%)	Post Empty Repositioning Workload	Change Percentage (%)
AIR	412,397	410,087	-0.56	411,073	-0.32
CPU	154,846	126,944	-18.02	129,750	-16.21
SPU	10,696	8,998	-15.88	10,843.9	1.38
D40	175	0	-100	0	-100
LC	290,981	223,232	-23.28	224,222	-22.94
MAR	5,876	3,649	-37.9	4,179	-28.88
MDV	0	0	0	0	0
PCA	1,111,500	1,089,640	-1.97	1,092,370	-1.72
PUP	27,657,600	29,262,600	5.8	29,360,400	6.16
Z53	6,142,480	4,163,890	-32.21	4,175,000	-32.03
REN	216	216	0	216	0
RR	80,035	68,619	-14.26	69,279	-13.44
Y	509,977	545,087	6.88	557,282	9.28
Z	800,648	696,273	-13.04	697,341	-12.9

CHAPTER 4
SHORT-TERM INVENTORY-AWARE FLEET MANAGEMENT IN SERVICE NETWORKS

4.1 Introduction

Operating a large ground service network involves, among others, ensuring that the right equipment is available at the right time at the right location. A fleet of different types of trailers and containers is used to transport freight between different locations. As demand is naturally imbalanced between regions, some facilities in the network will see more inbound than outbound trailers possibly leading to a build up of trailers that can exceed the facility capacity. Other facilities will see more outbound than inbound trailers possibly leading to equipment stock-outs and delays in executing planned freight movements. Having a heterogeneous fleet of equipment increases the complexity of equipment management as it destroys the self-balancing nature of driver circulations in the network, e.g., a driver can transport a 53-foot trailer from one location to another, but then return with two 28-foot trailers. Hours of service and union regulations may further complicate matters as it can result in (undesirable) bobtail movements, i.e., movements where a driver returns to his domicile in a tractor without pulling any trailer(s). To address equipment surplus or shortage at facilities, carriers reposition equipment – even using one-way rail movements – and lease equipment for short periods of time, all coming at a significant cost.

Effective equipment management requires short-term strategies to react to imbalances in the network as soon as they can be foreseen and long-term strategies that preemptively and proactively place equipment where it will likely be needed based on a demand forecast. At a long-term, tactical level where the planning horizon can span several months, a carrier focuses on equipment fleet size, e.g., whether expand or shrink the fleet, and redistributing the fleet across the network to

prepare for the future, e.g., the peak season, based on a demand forecast. Equipment leasing and procurement decisions are made a this level. These long-term tactical decisions are generally made infrequently (annually or bi-annually for major carriers). At a short-term, operational level where the planning horizon covers a few days, a carrier focuses on satisfying planned loads (planned movements) that are expected to be executed with high confidence in the upcoming days at least cost, possibly with equipment inventory level targets at facilities at the end of the planning horizon. In this case, accurate information about equipment inventory at facilities and in-transit equipment at the start of the planning period is critical. Short-term, operational equipment planning is the focus of our research.

The contributions of our research can be summarized as follows:

- We formulate a short-term inventory-aware ground equipment management problem. The input is a load plan and information about equipment inventory at facilities and in-transit equipment, and the output is a minimum cost assignment of equipment types to loaded movements and empty equipment repositioning movements;

- We present a complexity analysis for specific settings in terms of the number of equipment types;

- We introduce and time-expanded network formulation for the problem and propose a parsimonious time discretization scheme to control the size of the formulation;

- We develop an efficient and effective heuristic, which involves dynamically generating variables, for the solution of the formulation;

- We conduct an extensive computational study using large-scale instances provided by a major US carrier to assess the benefits of short-term inventory-aware ground equipment management and the efficacy of the proposed heuristic.

The remainder of this chapter is organized as follows. In Section 4.2, we briefly discuss relevant literature. In Section 4.3, we present a description of the problem and introduce a mixed integer programming formulation. In Section 4.5, we develop an efficient and effective heuristic for producing high-quality solutions. In Section 4.6, we give a summary of the results of an extensive computational study to assess the value of inventory-aware equipment management and the performance of the heuristic. Finally, in Section 4.7, we discuss future research directions.

4.2 Relevant literature

Equipment management in the trucking industry has been investigated from different perspectives in the literature. Fleet sizing, empty repositioning, and inventory control have been studied in both freight consolidation networks and small package networks and for different types of equipment (e.g., tractors, containers, trailers, etc.). These aspects are inter-connected, but researchers have studied them in isolation as well as in an integrated manner. To the best of our knowledge, there is no prior literature on inventory-aware equipment management with multiple substitutable equipment types. An early classification of empty equipment flow problems was presented by [36]. Multiple problem-defining characteristics are used, such as the type of flow (empty vs loaded movements), the transportation mode (single mode vs multi-mode), and the fleet homogeneity (single vs multiple substitutable equipment types). A more recent review of fleet planning problems [38] introduces a multi-modal fleet planning framework with a classification scheme based on problem and modeling characteristics and decision making levels. [39] analyze the relationship between fleet size and empty repositioning. Container planning in multi-modal transportation (especially rail and maritime modes) was studied by [40], [41], [33], and [37]. Trailer repositioning which is critical in so-called ground networks has been investigated by [34], [32]. Fleet heterogeneity was explored by [42] and [43]. Using equipment substitution to address equipment flow imbalance was studied by [44] for ground transportation and by [37] for maritime transportation; compatibility rules restrict the number and type of substitutions. Equipment heterogeneity natu-

rally occurs in other industries as well. In the airline industry, for example, most major carriers (e.g., American Airlines and Delta Airlines) operate different types of aircraft in different markets. A few airline carriers opt for a homogeneous fleet to simplify their operations (e.g., Southwest). [45] considers a heterogeneous airline fleet assignment problem. In the car rental industry, operating a heterogeneous fleet is crucial to be able to meet different customer preferences and brings many operational challenges. [46] surveys car rental literature and presents a conceptual framework of car rental fleet and revenue management.

Many equipment management problems can be modeled using time-expanded networks. However, the time-expanded networks quickly become prohibitively large and special solution techniques are required to solve them, e.g., column generation. An example of such an approach is [47] who consider a liner shipping cargo allocation problem.

In the United States, the heterogeneity in equipment types employed in the ground networks of less-than-truckload and package express carriers is mainly due to size. The three main equipment types are *short equipment* (also referred to as *pups*) with a typical length of under 28 feet, *long equipment* with length ranging from 40 to 48 feet, and *extra long equipment* with a typical length of 53 feet. Employing different size trailers improves utilization, reduces handling, and increases direct loading opportunities. Moreover, as a tractor can pull a combination of short equipment (typically two pups, but even three pups in some states) or a combination of long and short equipment, this allows loads that are bound to different locations to share a part of their route thereby reducing the number of driver schedules needed to execute loads.

4.3 Problem description

We consider the short-term planning of a fleet comprised of different types of trailers and containers for ground service network of a package express carrier. We are given a *load plan* for the planning period, typically a week. A load plan is the result of a load planning process that uses a demand forecast (and information on available resource types) to generate timed loads between pairs of

locations in the network and a tentative driver schedule to execute the planned loads. The loads are of three types: *loaded*, *empty*, and *bobtail* movements. The empty and bobtail loads present an initial step towards balancing equipment flows in the network. Each load has an associated set of compatible configurations of equipment types that can be assigned to it. Whether one configuration can be substituted by another configuration depends on multiple criteria, such as the size of the equipment, the existence of a pintle for short equipment (required to create a train of trailers), the ability of a facility to handle such equipment types, etc. These criteria can be used to create a substitution matrix that summarizes all the allowable equipment substitutions. During the load planning process a tentative equipment configuration is assigned to each load in the load plan. This tentative assignment is based on recently executed load plans in the hope that few adjustments are needed to account for week-to-week demand changes.

In addition to the load plan, we are given a snapshot of the equipment in the network at the start of the planning period (represented by time 0). This includes the inventory of equipment at every facility at time 0 and the in-transit (or en-route) equipment, i.e., equipment assigned to loads that were dispatched in the past (before time 0) and are expected to reach their destination before the end of the planning (represented by T). The inventory of equipment at the facilities (e.g., in the yard, undergoing maintenance, at a dock being loaded or unloaded) and the equipment assigned to in-transit loads represents the fleet of equipment available to execute the load plan.

Because the primary focus of load planning is ensuring capacity is available to move forecast demand and balancing equipment flow is only secondary consideration, If the load plan is executed as is, i.e., without changing the equipment configurations assigned to the loads or introducing additional empty equipment repositioning movements, equipment stock-outs may occur, which can cause delays in the delivery of demand and may be be costly to address at the time they occur. Our primary objective is to minimize the risk of equipment stock-out during the planning horizon (avoiding equipment stock-outs entirely is impossible because of unforeseen events that can happen during the planning period – equipment breakdowns, unexpected changes in demand,

etc.) either by changing the equipment configuration assigned to loads or by introducing empty equipment repositioning movements. A secondary objective may be to ensure a minimum target inventory of equipment types at facilities at the end of the planning period.

We will formulate a time-expanded network model for the problem outlined above in which nodes represent facility-time pairs and arcs represent planned timed loads in the load plan or potential empty equipment repositioning movements.

Next, we summarize the notation that we adopt to describe the model and the proposed solution approach. After that, we present a mixed integer programming formulation for the problem.

4.3.1 Notation

The following parameters are used in the definition of the problem and its mixed integer programming formulation:

- $\mathcal{F}$: The set of facilities in the network.
- $\mathcal{E}$: The set of equipment types. These can differ by size, i.e, short (trailers with a length of less than or equal to 28 feet), long (trailers with a length ranging from 40 to 48 feet), and extra long (trailers with a length of 53 feet). They can also differ by utility, e.g., refrigerated or heated trailer, rail containers, etc.,
- $\mathcal{C}$: The set of equipment configurations. Each configuration is a vector representing a possible combination of units of equipment types in $\mathcal{E}$. Some configurations are only allowed in certain regions. For example, configurations containing three pups are allowed in only 13 states. Let η denote the configuration matrix where rows represent configurations in $\mathcal{C}$ and columns represent equipment types in $\mathcal{E}$, then an entry η_{ce} represents the number of units of equipment type e in configuration c. An example of η with three equipment types in $\mathcal{E}$ and

four configurations in $\mathcal{C}$ is shown below:

$$\eta = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{array} \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \end{array}$$

In this example, configuration c_2 represents a two-unit train of short equipment of type e_1.

- $\mathcal{L}$: The set of timed loads scheduled to dispatch within the planning period $[0, T]$. A load captures the time and the location where a trailer is to be loaded and the time and the location where it is to be unloaded. A load $l \in \mathcal{L}$ has the following attributes:
  - $o(l) \in \mathcal{F}$: The origin of load l.
  - $t^o(l) \in [0, T]$: The time at which equipment starts being loaded at the origin. Let $\mathcal{T}^o(i)$ denote the set of times at which equipment starts being loaded at terminal i.
  - $d(l) \in \mathcal{F}$: The destination of load l.
  - $t^d(l) \in [0, T]$: The time at which equipment ends being unloaded at the destination. Let $\mathcal{T}^d(i)$ denote the set of times at which equipment ends being unloaded at terminal i.
  - $q(l) \in \mathcal{C}$: The (initial) equipment configuration assigned to load l.
  - $S_l \subseteq \mathcal{C}$: The set of allowable configurations for load l.
- $\mathcal{N}$: The set of nodes in the time-expanded network. Each node (i, t) in $\mathcal{N}$ represents a facility $i \in \mathcal{F}$ and a time $t \in \mathcal{T}(i)$ with $\mathcal{T}(i)$ representing the set of times for facility i, i.e., $\mathcal{T}(i) = \{t : (i, t) \in \mathcal{N}\}$. The set $\mathcal{T}(i)$ contains times 0 and T and all other $t \in \mathcal{T}(i)$ have $0 < t < T$.

For convenience, for time point $t \in \mathcal{T}(i) \setminus \{0, T\}$, we let $t^- = \max\{s \in \mathcal{T}(i) \ : \ s < t\}$ be the preceding time point and $t^+ = \min\{s \in \mathcal{T}(i) \ : \ s > t\}$ be the succeeding time point (and, thus, (i, t^-) and (i, t^+) represent, respectively, the node preceding and the node succeeding (i, t)).

We also define the sets $\mathcal{L}^-_{(i,t)}$ and $\mathcal{L}^+_{(i,t)}$ as the sets of inbound and outbound loads in $\mathcal{L}$ associated with node (i, t) in $\mathcal{N}$, respectively:

$$\mathcal{L}^-_{(i,t)} = \{l \in \mathcal{L} : \ d(l) = i, \ t^- \leq t^o(l) < t\},$$
$$\mathcal{L}^+_{(i,t)} = \{l \in \mathcal{L} : \ o(l) = i, \ t^- < t^d(l) \leq t\}.$$

- $\mathcal{A}$: The set of arcs linking nodes in $\mathcal{N}$. An arc a linking two nodes (i, t_1) and (j, t_2), represents the possibility of sending empty equipment from facility i at time t_1 and making it available at facility j by time t_2. For a given node $(i, t) \in \mathcal{N}$, we define the sets $\delta^-_{(i,t)}$ and $\delta^+_{(i,t)}$ as the sets of arcs in $\mathcal{A}$ that are inbound and outbound to (i, t) respectively.

- I_{ie}: The inventory of equipment type e at facility i at the start of the planning horizon.

4.3.2 Model

We present a mixed integer programming formulation of the inventory-aware equipment management model. At time 0, each facility i in $\mathcal{F}$ has an initial inventory I_{ie} of equipment type e in $\mathcal{E}$. If the load plan were to be executed without any adjustments, it is possible that the inventory of some equipment type drops below zero during the planning period. Our objective is to prevent this from happening by adjusting the load plan in one of two ways (or both):

1. **Equipment substitution**: assigning different equipment configurations (from the set of eligible equipment configurations) to loads.

2. **Empty repositioning**: adding one or more empty equipment repositioning movements between pairs of facilities to redistribute equipment from locations where there is a surplus to places where there is a shortage of a given equipment type. The judicious timing of any empty equipment repositioning movements is critical.

Equipment substitution and empty repositioning decisions incur costs for carriers. We ignore equipment substitution costs as they are negligible compared to empty equipment repositioning costs. The optimization model seeks to minimize the transportation costs of any added empty equipment repositioning movements. The solution to the optimization model needs to satisfy the following constraints:

1. **Load equipment substitution**: every planned load l can be assigned exactly one equipment configuration in S_l,

2. **Inventory flow balance**: at every facility, the inventory of a given equipment type is monitored during the planning period; properly accounting for arriving and departing loads and any added empty equipment repositioning movements,

3. **Non-negative inventory**: to prevent any equipment stock-out, inventory is not allowed to drop below zero during the planning period. This constraint can be generalized to take safety stock considerations into account. Incorporating safety stock can help protect against execution uncertainty (e.g., load and unload times, load cancellation, ad-hoc movements, transit times, etc.). It is also possible to incorporate inventory limits at facilities, e.g., capturing limited yard space.

4. **Target inventory**: planners may or may not require a certain inventory of equipment at a facility at the end of the planning period. Inventory targets can be used to better position the system for anticipated future load demand.

Decision variables

- s_{iet}: inventory of equipment type $e \in \mathcal{E}$ at node $(i,t) \in \mathcal{N}$,
- y_{lc}: whether or not equipment configuration $c \in S_l$ is assigned to load $l \in \mathcal{L}$,
- u_{ae}: number of repositioning movements of equipment type $e \in \mathcal{E}$ added on arc $a \in \mathcal{A}$.

Formulation

$$(\mathcal{IAM}) \quad \min \sum_{a \in \mathcal{A}} \sum_{e \in \mathcal{E}} D_{ae} u_{ae} \tag{4.1}$$

$$\text{s.t. } s_{ie0} = I_{ie}, \quad i \in \mathcal{F}, e \in \mathcal{E}, \tag{4.2}$$

$$s_{iet} = s_{iet^-} + \left(\sum_{l \in \mathcal{L}^-_{(i,t)}} \sum_{c \in S_l} \eta_{ce} y_{lc} - \sum_{l \in \mathcal{L}^+_{(i,t)}} \sum_{c \in S_l} \eta_{ce} y_{lc} \right) + \left(\sum_{a \in \delta^-(i,t)} u_{ae} - \sum_{a \in \delta^+(i,t)} u_{ae} \right), \ (i,t) \in \mathcal{N}, \ t > 0, \ e \in \mathcal{E}, \tag{4.3}$$

$$\sum_{c \in S_l} y_{lc} = 1, \quad l \in \mathcal{L}, \tag{4.4}$$

$$y_{lc} \in \{0,1\}, \quad l \in \mathcal{L}, \ c \in S_l, \tag{4.5}$$

$$s_{iet} \in \mathbb{Z}_{\geq 0}, \quad (i,t) \in \mathcal{N}, e \in \mathcal{E}, \tag{4.6}$$

$$u_{ae} \in \mathbb{Z}_{\geq 0}, \quad a \in \mathcal{A}, \ e \in \mathcal{E}, \tag{4.7}$$

where D_{ae} represents the cost of executing an empty movement with equipment type e in $\mathcal{E}$, on arc a in $\mathcal{A}$. For simplicity, we use the distance of arc a to represent the cost.

The objective function (4.1) represents the transportation costs of all the new empty movements generated by the model. Constraints (4.2) set the initial inventory. Constraints (4.3) ensure flow balance at each node $(i,t) \in \mathcal{N}$ for each equipment type $e \in \mathcal{E}$. Constraints (4.4) ensure that every

load l is assigned exactly one configuration in the set S_l.

Target inventories at the end of the planning period can be accommodated by adding lower and upper bounds $\underline{s}_{ie}^T$ and $\bar{s}_{ie}^T$ on the variables s_{ieT}, i.e.,

$$\underline{s}_{ie}^T \leq s_{ieT} \leq \bar{s}_{ie}^T. \tag{4.8}$$

4.4 Complexity Results

We analyze the complexity of the proposed inventory-aware model based on the configuration matrix η. In service networks with a homogeneous fleet equipment management, i.e., empty repositioning of equipment, can be modeled as a single commodity network flow problem and is therefore solvable in polynomial time. When the fleet is heterogeneous, however, equipment management becomes more difficult. This has been shown formally in [44], which presents a complexity analysis of equipment balancing in a flat network with multiple equipment types. The problem becomes NP-hard when the fleet is comprised of three or more equipment types. For this problem, we consider configuration matrices where rows have entries 0 and 1 only (i.e, we don't allow composite configurations where there is more than one unit of an equipment type, such as the two-pup train configuration that is commonly used in North America). For this case, we present a problem relaxation where we keep the network flow structure, and thus can be solved as a minimum cost flow problem. Finally, we analyze a theoretical case with target inventory that can be proven to be NP-hard.

4.4.1 Single-equipment configuration case

We consider the case where the configuration matrix η has only one entry 1 and all remaining entries equal to zero for each row (i.e., the rows of the configuration matrix are a subset of the rows of the identity matrix of size $|\mathcal{E}|$). This means that each eligible configuration is comprised of exactly one single unit of a specific equipment category e in $\mathcal{E}$ and no composite configurations

(train of at least two units of equipment) are allowed, as in this example:

$$\eta = \begin{array}{c} \\ c_1 \\ c_2 \\ c_2 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

In this example, configurations c_1, c_2, c_3 are made of exactly one single unit of equipment categories e_1, e_3, and e_4 respectively.

Network Flow Relaxation .

Using the same type of configuration matrices with single equipment, we consider a relaxation of the model that keeps the Network Flow structure without completely removing constraints (4.4) and thus formulating it as a minimum-cost flow problem. We create a time-expanded network with nodes (i, e, t) where i is a facility in $\mathcal{F}$, e is an equipment type in $\mathcal{E}$ and t is a discrete time in $[0, T]$. For a given facility equipment pair (i, e), we add arcs linking nodes (i, e, t) and $(i, e, t + 1)$ and associate flow variables s_{iet} to them. We also add arcs between nodes (i, e, t_1) and (j, e, t_2) that represent the possibility of sending empty units of equipment e from facility i at time t_1 and making it available at facility j at time t_2. We associate flow variables u_{ae} to them. For every equipment type e, we add a source node S_e and a sink node T_e. We add arcs from S_e to nodes $(i, e, 1)$ with capacity I_{ie0} (initial inventory), and from nodes (i, e, n_T) to T_e. S_e sends $\sum_{i\in\mathcal{F}} I_{ie0}$ units of flow to T_e. For every load $l \in \mathcal{L}$, we add a source node s_l and a sink node t_l. We add arcs with capacity 1 from s_l to nodes $(d(l), e, t^a(l)$ and from nodes $(o(l), e, t^d(l)$ to t_l. We add a super source node S and super sink node T. We add arcs of capacity 1 from S to nodes s_l and from nodes t_l to T, and arcs from S to nodes S_e and from nodes T_e to T.

With this network flow representation, the initial model is equivalent to solving a minimum cost

flow problem where we send $\sum_{e\in\mathcal{E}}\sum_{i\in\mathcal{F}} I_{ie0} + |\mathcal{L}|$ units of flow from S to T. An illustration is given in figure 4.1.

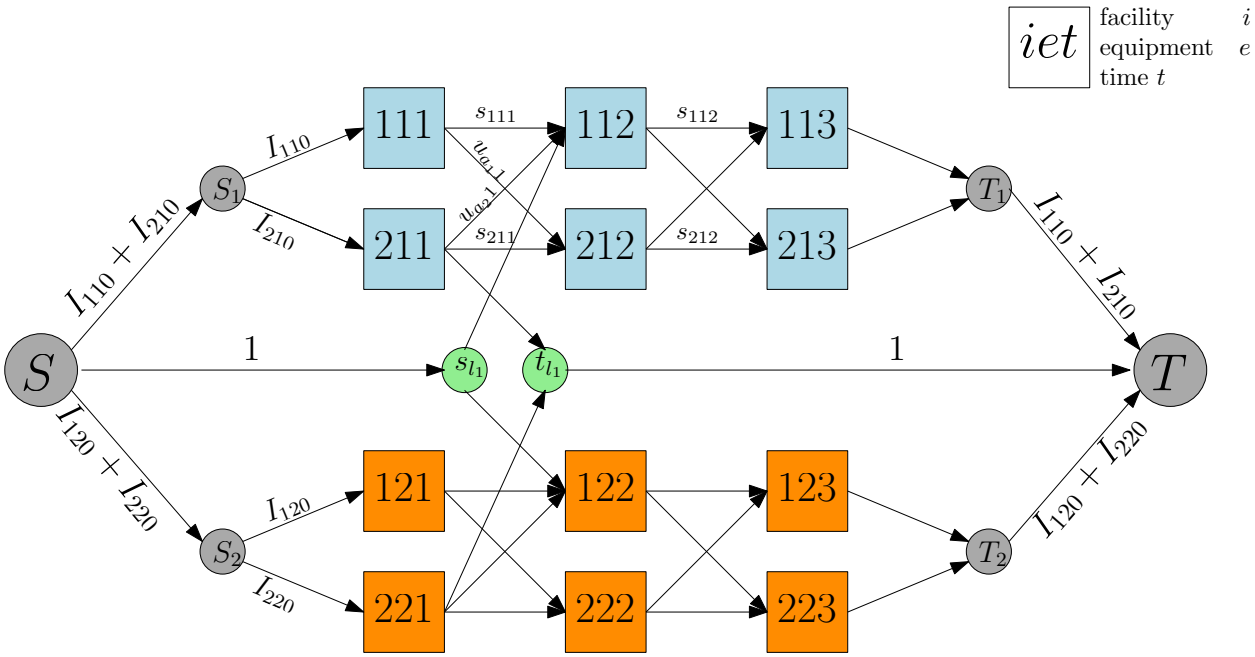


Figure 4.1: Illustration of a flow structure with two equipment types. Load l_1 departs from facility 2 at time 1 and arrives at facility 1 at time 2. Only one planned load is represented for illustration.

When the target inventory constraints are enforced, the problem remains polynomially solvable. It suffices to add the flow bounds $\underline{s}^T_{ie}$ and $\bar{s}^T_{ie}$ to the arcs linking nodes (i, e, n_T) to T_e for each facility-equipment pair (i, e).

4.4.2 One-to-many substitutions with target inventories

We consider the general case where the configuration matrix contains more than two rows and the model needs to satisfy a target inventory at the end of the horizon. This case can be proven to be difficult to solve through the following proposition:

Proposition 1. *The problem of finding a feasible assignment of equipment configurations to loads to satisfy a non-negative inventory throughout the horizon and a final target inventory is NP-complete.*

Proof. Transformation from PARTITION PROBLEM, which is known to be NP-complete.

PARTITION PROBLEM: Given a set of positive integer variables $\mathcal{S} = \{a_1, a_2, ..., a_n\}$, can we partition it into two subsets S_1 and S_2 such that the sum of the numbers in $\mathcal{S}_1$ is equal to the sum of numbers in $\mathcal{S}_2$?

We create one instance of the inventory-aware model with two facilities $\mathcal{F} = \{1, 2\}$ and one equipment category $\mathcal{A} = \{e_1\}$. We use the following configuration matrix with $n + 1$ rows:

$$\eta = \begin{array}{c} \\ c_1 \\ c_2 \\ . \\ c_{n-1} \\ c_n \\ c_{n+1} \end{array} \begin{array}{c} e_1 \\ \left[\begin{array}{c} a_1 \\ a_2 \\ . \\ a_{n-1} \\ a_n \\ 0 \end{array}\right] \end{array}$$

We consider a time horizon $[0, 2]$ and a set of loads $\mathcal{L} = \{l_1, l_2, ..., l_n\}$. Each load l_i departs from facility 1 at time 1, arrives at facility 2 at time 2, and the set of its eligible equipment configurations is $S_{l_i} = \{c_i, c_{n+1}\}$. The initial inventories at facilities 1 and 2 are, respectively, $I_{10} = \sum_{i=1}^{n} a_i$ and $I_{20} = 0$. the target inventories are $s_{12} = s_{22} = \sum_{i=1}^{n} a_i/2$.

For each load l_i, we assign a substitution variable y_{l_i} that is binary such that:

$$y_{l_i} = \begin{cases} 1, & \text{if configuration } c_i \text{ is used on load } l_i, \\ 0, & \text{otherwise.} \end{cases}$$

We create a time expanded network with three nodes $\mathcal{N} = \{(1, 1), (1, 2), (2, 2)$ and we assume the set of empty repositioning arcs is empty. The set of inventory flow constraints can be written as

follows:

$$
\begin{aligned}
s_{11} &= I_{10} - \sum_{i=1}^{n} a_i y_{l_i} \\
s_{12} &= s_{11} \\
s_{22} &= I_{20} + \sum_{i=1}^{n} a_i y_{l_i} \\
s_{12} &= \sum_{i=1}^{n} a_i/2 \\
s_{22} &= \sum_{i=1}^{n} a_i/2
\end{aligned}
\tag{4.9}
$$

This entails that the model is feasible *if and only if*:

$$\sum_{i=1}^{n} a_i y_{l_i} = \sum_{i=1}^{n} a_i/2$$

This proves that a Yes-instance of PARTITION PROBLEM yields a feasible instance of the inventory-aware model and vice-versa. ■

4.5 Methodology

Instances of the formulation for the short-term inventory-aware ground equipment management problem tend to be very difficult to solve. The main reason is that the size of the instances for the service networks of interest becomes prohibitively large. The number of facilities, the number of equipment configurations, and the number of loads is already very large, but the number of possible empty equipment repositioning movements is astronomical for a fine discretization of time (the number is of the order of $0.5 \times (n \times t)^2 \times e$ with n the number of facilities, t the number of time points

(at a facility), and e the number of equipment configurations, e.g., a week-long planning period with an hourly discretization of time would results in the order of $(1500 \times 168)^2 \times 20 \approx 0.64 \times 10^{12}$ possible empty equipment repositioning movements).

In this section, we explore approaches to solve instances of the the formulation in a reasonable amount of time by judiciously choosing a discretization of time and generating empty equipment repositioning movement options dynamically.

4.5.1 Time discretization

The time discretization, i.e., the choice of the sets $\mathcal{T}(i)$ for $i \in \mathcal{F}$ is an essential feature of the inventory-aware equipment management problem and affects two aspects of the model. First, the inventory of equipment at facilities needs to be evaluated at certain time points to avoid (or minimize) the risk of equipment stock outs. The larger the number of time points, the smaller the risk of stock-outs (as a stock-out can only occur between two consecutive time points), but the larger the number of time points, the larger the formulation. Second, the set of time points at a facility defines the possible departure times for empty equipment repositioning movements. The larger the number of time points, the more empty equipment repositioning movement options, but the larger the number of time points, the larger the formulation. Thus, the choice of time points is critical when seeking to find high-quality solution in a reasonable amount of time. Finally, it is important to recognize that the times at which you evaluate equipment inventory at a facility and the times at which you consider dispatching empty equipment to another facility do *not* have to be the same.

We focus first on the set of times points at a facility at which we will evaluate the equipment inventory. Our approach is motivated by the fact that a stock-out only occurs at a time when a load departs, i.e., the load requires a certain equipment type, but the inventory of that equipment type at the facility is zero. This implies that evaluating equipment inventory at every load departure suffices to identify stock-outs, if any. However, we can do even better. At each facility i, we

aggregate inbound and outbound loads in $\mathcal{L}$ into inbound and outbound blocks such that within each inbound block of loads there is no outbound load, and within each outbound block of loads there is no inbound load. Let the set of nodes of the time-expanded network, $\mathcal{N}$, be formed by pairs (i, t) with t the start loading time $t^o(l^\star)$ of the last load $l^\star$ in each outbound block at facility i. (In the worst case, this implies a node for every departing load, i.e., $|\mathcal{N}| = |\mathcal{L}|$.) Figure 4.2 depicts an example of this aggregation. In the example, the set of time points at the terminal will

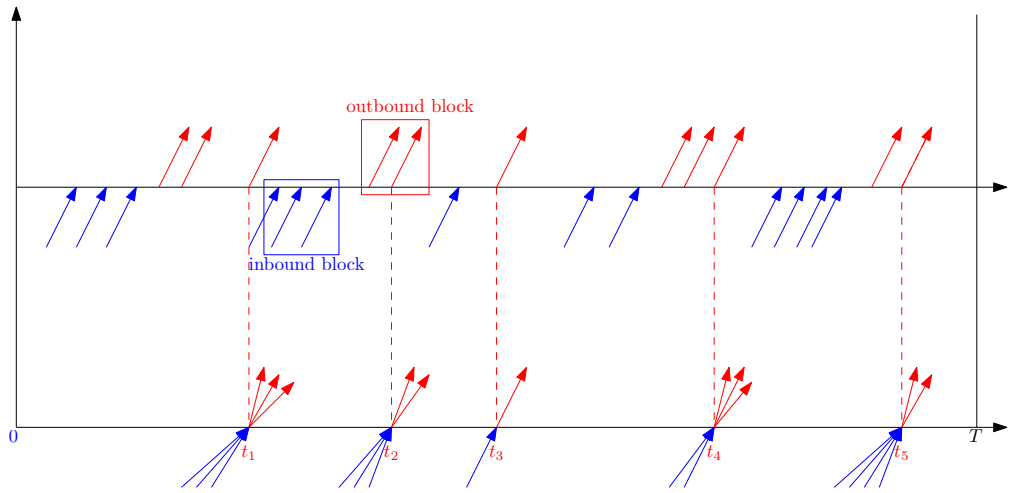


Figure 4.2: Example of inbound and outbound blocks in a given terminal

be $\{0, t_1, t_2, t_3, t_4, t_5, T\}$ and the set of nodes in the time-expanded network for the terminal will be $\{(i, 0), (i, t_1), (i, t_2), (i, t_3), (i, t_4), (i, t_5), (i, T)\}$. Next, We formally prove the validity of the aggregation scheme.

Proposition 2. *For a given facility-equipment type pair, there will be no equipment stock-out during the planning period $[0, T]$ if and only if the equipment inventory is non-negative at the start of the planning period and at the end of each outbound block.*

Proof. For a given facility i and equipment type e, let $Inv_{ie} : t \mapsto Inv_{ie}(t)$ denote the function that monitors the inventory of equipment type e at any time t in $[0, T]$. We want to prove that Inv_{ie}

is a nonnegative function *if and only if* it is nonnegative at the time points in $\mathcal{T}(i)$, i.e.:

$$\forall t \in [0,T] \quad Inv_{ie}(t) \geq 0 \iff \forall t \in \mathcal{T}(i) \quad Inv_{ie}(t) \geq 0$$

The direction $\implies$ is trivial as any time-point t in $\mathcal{T}(i)$ is in $[0,T]$. For the direction $\impliedby$ let $\mathcal{T}(i) = \{0 = t_1, t_2, \ldots, t_{|\mathcal{T}(i)|} = T\}$. We consider two cases:

- $t \in [t_j, t_{j+1}]$ with $j \leq |\mathcal{T}(i)| - 1$: By the definition of a block, in the interval $[t_j, t_{j+1}]$ there will first be a set of inbound loads (possibly empty), followed by a set of outbound loads (possibly empty). Thus, there is a unique time point, t^M, at which the maximum inventory during the interval is reached for the first time. Hence, if $t \in [t_j, t_j^M]$ then $Inv_{ie}(t) \geq Inv_{ie}(t_j) \geq 0$, and if $t \in [t_j^M, t_{j+1}]$ then $Inv_{ie}(t) \geq Inv_{ie}(t_{j+1}) \geq 0$.

- $t \in [t_{|\mathcal{T}(i)|-1}, t_{|\mathcal{T}(i)|}]$: After $t_{|\mathcal{T}(i)|-1}$ there are only arriving loads. Hence, the inventory only increases after $t_{|\mathcal{T}(i)|-1}$. Thus, we have $Inv_{ie}(t) \geq Inv_{ie}(t_{|\mathcal{T}(i)|-1}) \geq 0$.

■

Although this aggregation scheme ensures that stock-outs can be avoided, it may have two undesirable features. First, at busy facilities with many daily inbound and outbound loads, the aggregation scheme may generate many time points with little time separation due to many alternating small inbound and outbound blocks. Second, at less busy facilities with few daily inbound and outbound loads or with more inbound than outbound or more outbound than inbound loads, this aggregation scheme may generate few time points. Enforcing no stock-outs at a facility between two consecutive time points that are close in time may be unnecessary and having only a few time points at a facility may prevent necessary empty equipment repositioning movements. To address these issues, at busy facilities we enforce a minimum time separation between time points (τ_m) at which we enforce positive inventory and at less busy facilities we enforce a maximum time separation between time points (τ_M), by adding additional time points if necessary, to ensure sufficient opportunities for empty equipment repositioning.

4.5.2 Solving the LP relaxation

Even with a judicious choice of time points at facilities, for a large ground service network, the set of possible empty equipment repositioning arcs, $\mathcal{A}$, can be prohibitively large. Including all repositioning arcs in the formulation may result in memory issues or excessive solution times, even for just solving the LP relaxation. Moreover, only a few of the repositioning arcs will likely be chosen in an optimal solution (i.e., only a few additional empty equipment repositioning movements will be introduced). Therefore, we generate repositioning arc variables dynamically as needed, i.e., we use a column generation approach to solve the LP relaxation. To be able to define the reduced cost of a repositioning arc variable given the solution to a restricted formulation (i.e., a formulation in which many repositioning arc variables have been ommitted), we need to look at the dual of the LP relaxation. Let the dual variables associated with Constraints 4.3, 4.4, and 4.5 of the LP relaxation of $\mathcal{IAM}$ be π_{iet}, β_l, and γ_{lc}, respectively. Then the dual problem is

$$(\mathcal{D}-\mathcal{IAM}) \quad \max \sum_{(i,0)\in\mathcal{N}} \sum_{e\in\mathcal{E}} I_{ie}\pi_{ie1} - \sum_{l\in\mathcal{L}} \left(\beta_l + \sum_{c\in S_l} \gamma_{lc} \right) \tag{4.10}$$

$$\text{s.t. } \pi_{iet} - \pi_{iet^+} \leq 0,\ \forall (i,t)\in\mathcal{N},\ 0<t<T,\ e\in\mathcal{E}, \tag{4.11}$$

$$\pi_{ieT} \leq 0,\ \forall (i,T)\in\mathcal{N},\ e\in\mathcal{E}, \tag{4.12}$$

$$\pi_{iet} - \pi_{jet'} \leq D_{ae},\ \forall a=((i,t)\rightarrow(j,t'))\in\mathcal{A},\ e\in\mathcal{E}, \tag{4.13}$$

$$\eta_{ce}(\pi_{iet} - \pi_{jet'}) - \beta_l - \gamma_{lc} \leq 0, \quad l=((i,t)\rightarrow(j,t'))\in\mathcal{L},\ c\in S_l, \tag{4.14}$$

$$\gamma_{lc} \geq 0, \quad l\in\mathcal{L},\ c\in S_l, \tag{4.15}$$

$$\pi_{iet}, \beta_l \text{ free} \quad (i,t)\in\mathcal{N}, t>0,\ e\in\mathcal{E}. \tag{4.16}$$

Observe that Constraints (4.11) and (4.12) imply that the dual variables π_{iet} are non-positive and monotonically non-decreasing with respect to t. This observation will be used to speed up the dynamic variable generation strategy.

Next, assume that we have a solution to a restricted LP relaxation that only includes a subset $\mathcal{A}_1 \subseteq \mathcal{A}$ of repositioning arcs, then finding a variable $u_{a'e'}$ with $a' \in \mathcal{A} \setminus \mathcal{A}_1$ and $e' \in \mathcal{E}$ with minimum reduced cost amounts to solving pricing problem:

$$\min_{\substack{e\in\mathcal{E},\, a\in\mathcal{A}\setminus\mathcal{A}_1 \\ a=((i,t),(j,t'))}} D_{ae} - \pi_{iet} + \pi_{jet'} \tag{4.17}$$

If the minimum is non-negative, then the solution to the restricted LP relaxation is also optimal to the (full) LP relaxation. Otherwise, we have identified a variable that should be added to the restricted LP relaxation.

Adding one variable at a time, however, is computationally too expensive as it will require the solution of many (still large) restricted LP relaxations. Therefore, instead, we search for and add a number of negative reduced cost variables in each iteration. This results in the following algorithm for solving the LP relaxation of $\mathcal{IAM}$, where parameter N_{iter} indicates the maximum number of negative reduced cost variables that are generated and added to the restricted LP relaxation in a single iteration:

- **Step 0**: Initialize $\mathcal{A}_1$ with a small subset of repositioning arc variables,
- **Step 1**: Solve the restricted LP relaxation with subset $\mathcal{A}_1$,
- **Step 2**: Generate a set $\mathcal{A}_2 \subseteq \mathcal{A} \setminus \mathcal{A}_1$ of up to N_{iter} negative reduced cost arc repositioning variables. If $\mathcal{A}_2 = \emptyset$, go to **Step 4**,
- **Step 3**: Add the columns in $\mathcal{A}_2$ to $\mathcal{A}_1$. Go to **Step 1**,
- **Step 4**: Stop. An optimal solution to the LP relaxation has been found.

To generate negative reduced cost repositioning arc variables (in **Step 2**), we consider three strategies : BASIC, a simple enumeration strategy, ENHANCED BASIC, a more intelligent enumer-

ation strategy that favors diversity, and EFFICIENT ENHANCED BASIC - a sophisticated enumeration strategy that exploits dual information to guide and restrict the search.

BASIC STRATEGY Our naive enumeration strategy iterates over equipment types and facilities in no particular order. For each combination of equipment type e and facility i, it iterates over the set of facilities that can reach facility i directly, i.e., its inbound arcs, again in no particular order, and for each outbound arc, iterates over the time points in $\mathcal{T}(i)$. If the reduced cost of the associated repositioning arc variable is negative, it is added to the set $\mathcal{A}_2$. The enumeration stops as soon as N_{iter} negative reduced cost variables have been found. The exact same search is performed in each iteration.

ENHANCED BASIC STRATEGY To introduce more diversity in the set of generated negative reduced cost repositioning arc variables, we impose limits on the number of negative reduced cost variables generated for each equipment type e, N_e, for each facility i, N_f, and for each outbound arc, N_a. Furthermore, when sorting is enabled, we iterate over the equipment types and the facilities in a certain order to increase the chances of finding negative reduced cost variables early in the enumeration. We iterate over the equipment types $e \in \mathcal{E}$ in nonincreasing order of

$$\lambda_e = \frac{\textit{\# explored variables with negative reduced cost}}{\textit{\# explored variables}},$$

where λ_e is computed based on information gathered in the previous iteration. Similarly, within each equipment type e, we iterate over the facilities in nonincreasing order of λ_{ei}, defined similar to the quantity λ_e at the facility level. In the first iteration, we set $\lambda_e = 1$ for $e \in \mathcal{E}$ and $\lambda_{ei} = 1$ for $e \in \mathcal{E}, i \in \mathcal{F}$. When sorting is disabled, we use a round robin scheme that works as follows. In each iteration, we start from the last equipment type explored in the previous iteration, and for each equipment type, we start from the last facility explored in the previous iteration.

Moreover, when sorting is enabled, we truncate the search of equipment categories using the

λ_e values. Specifically, we stop the enumeration as soon as we reach an equipment category with $\lambda_e = 0$, provided that a minimum number of negative reduced cost variables were found earlier in the iteration. The rationale for this heuristic idea is as follows. If no negative reduced cost variables were found for an equipment type in the previous iteration, i.e., no empty repositioning of equipment appeared advantageous, it is likely that no negative reduced cost variables will be found in the current iteration, and searching for them may be a waste of time. This idea is especially useful in practice, as companies often have large number of equipment types, often more than ten, but primarily use a few, often only three or four. To ensure the linear program is solved to optimality, we do not stop the search early when no negative reduced cost variables have been found up to that point.

In addition to the control parameters N_{iter}, N_e, N_f and N_a, we use the following additional parameters:

$Sort$: A boolean that when set to true activates the sorting of sets when searching for columns with negative reduced cost. Equipment categories and facilities are processed based on the order explained earlier. When set to false, a round robin scheme is used to diversify the processing of equipment types and facilities.

$Best$: A boolean that when set to true selects the N_a most negative reduced cost timed repositioning arcs (i.e., for a pair of facilities), and when set to false selects the first N_a negative reduced cost repositioning arcs.

l, m : These quantities as associated with the round robin scheme. l represents the index of the last equipment type explored in the previous iteration, and m represents a list of indexes of the last facilities (one for each equipment type) explored in the previous iteration.

Algorithm 10 gives the pseudo-code for this strategy.

EFFICIENT ENHANCED BASIC STRATEGY The previous strategies may unnecessarily evaluate

Algorithm 10: ENHANCED-BASIC(N_{iter},N_e,N_f,N_a,$Sort$,$Best$,ℓ,m,λ)

$\mathcal{F}_1, \mathcal{E}_1 \leftarrow$ *unordered lists of facilities in the network and equipment categories*
$\mathcal{C} \leftarrow \{\}$
$k_1 \leftarrow \ell + 1$ *// index of last equipment category searched in previous iteration*
if $Sort$ **then**
  $\mathcal{E}_1 \leftarrow$ *Equipment categories sorted by* λ *in non-increasing order*
  $k_1 \leftarrow 1$
for ***each*** *equipment type* ***e*** $= k_1, .., |\mathcal{E}_1|$ *in* $\mathcal{E}_1$ **do**
  $\mathcal{C}_e \leftarrow \{\}$
  $k_2 \leftarrow m_e + 1$ *// index of last facility searched in previous iteration for equipment type* e
  if $Sort$ **then**
    $Facilities \leftarrow$ *facilities sorted by* λ *in non-increasing order*
    $k_2 \leftarrow 1$
  for ***each*** *facility* ***i*** $= k_2, .., |\mathcal{F}_1|$ *in* $\mathcal{F}_1$ **do**
    $\mathcal{T}(i) \leftarrow$ *set of time-points at facility* i *in the order of time*
    $Inbound[i] \leftarrow$ *unordered list of facilities* j *with arc* (j, i)
    $\mathcal{C}_f \leftarrow \{\}$
    for ***each*** *facility* ***j*** *in* $Inbound[i]$ **do**
      $\mathcal{C}_a \leftarrow \{\}$
      for ***each*** *time-point* ***t*** *in* $\mathcal{T}(i)$ **do**
        $a = (j, t_j) \rightarrow (i, t)$ *// available empty repositioning arc*
        if $D_{ae} + \pi_{iet} - \pi_{jet_j} < 0$ **then**
          $\mathcal{C}_a \leftarrow a$
        if $|\mathcal{C}_a| \geq N_a$ **then**
          break
      $\mathcal{C}_f \leftarrow \mathcal{C}_f \cup \mathcal{C}_a$
      if $|\mathcal{C}_f| \geq N_f$ **then**
        break
    $\mathcal{C}_e \leftarrow \mathcal{C}_e \cup \mathcal{C}_f$
    if $|\mathcal{C}_e| \geq N_e$ **then**
      break
  if $|\mathcal{C}| > 0$ & $\lambda_{e+1} = 0$ **then**
    break
  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_e$
  if $|\mathcal{C}| \geq N_{iter}$ **then**
    break
return $\mathcal{C}$

the reduced cost of many repositioning arc variables. By cleverly exploiting dual information, such evaluations can be avoided, which will improve the efficiency. Furthermore, exploiting dual information may also lead to more effective evaluation orders (e.g., the order in which facilities are examined). For each combination of equipment type e and facility i, the dual variables π_{iet} are nonpositive and monotonically nondecreasing in $t \in \mathcal{T}(i)$, i.e.,

$$\pi_{iet_1} \leq \pi_{iet_2} \leq \cdots \leq \pi_{ieT} \leq 0. \tag{4.18}$$

This follows from Constraints (4.11) and (4.12) in dual formulation $\mathcal{D} - \mathcal{IAM}$.

This property can be exploited to avoid enumerating (some) repositioning arc variables. For a given equipment type e and repositioning arc $a = ((j, t'), (i, t))$, the reduced cost $D_{ae} + \pi_{iet} - \pi_{jet'}$ can be divided into parts π_{iet} and $D_{ae} - \pi_{jet'}$. As π is nonpositive, the first part is always nonpositive and the second part is always nonnegative.

By using appropriate orderings, we can stop the enumeration early in three situations. First, suppose that for a given equipment type e the facilities are given in non-decreasing order of

$$\underline{\pi}_{ei} = \min_{t \in \mathcal{T}(i)} \pi_{iet}.$$

Then, we can stop the enumeration as soon as we reach a facility with $\underline{\pi}_{ei} = 0$, as the reduced costs for all repositioning arc variables for all remaining facilities will be nonnegative. Second, suppose that for a given combination of equipment e type and facility i, the inbound arcs (j, i) are given in nondecreasing order of $D_{(ji)e} - \bar{\pi}_{ej}$ with

$$\bar{\pi}_{ej} = \max_{t \in \mathcal{T}(j)} \pi_{jet}.$$

Then, we can stop the enumeration as soon as we reach an inbound arc with $\underline{\pi}_{ei} + D_{(ji)e} - \bar{\pi}_{ej} > 0$. Finally, for a given inbound arc (j, i), because we enumerate time points in increasing order of

time, we can stop the enumeration as soon as we reach a repositioning arc with $D_{(ji)e} + \pi_{iet} \geq 0$.

Exploiting dual information as described does require sorting and thus comes at a price, but hopefully the time spent in sorting is offset by far fewer reduced cost evaluations. The effect of the $Sort$ parameter is redefined as follows in this variant:

$Sort$: A boolean that when set to true activates the sorting of sets when searching for columns with negative reduced cost. Equipment categories are processed in nonincreasing order of their contribution to the objective function in the last iteration. For a given equipment category e, facilities are processed in non-decreasing order of $\underline{\pi}_{ie}$. For a given facility i, the inbound arcs (j, i) are processed in non-decreasing order of $D_{(ji)e} - \bar{\pi}_{ej}$. When set to false, a round robin scheme is used to diversify the processing of equipment types and facilities.

Algorithm 11 gives the pseudo-code for this strategy.

Each of the three pricing algorithms discussed above, i.e., BASIC, ENHANCED-BASIC, and EFFICIENT-ENHANCED-BASIC, can be embedded in the iterative algorithm LP-HEUR for approximately solving the LP relaxation of $\mathcal{IAM}$ outlined in Algorithm 12. LP-HEUR uses three additional parameters, K_1, K_2, and N_{LP}. Parameters K_1 and K_2 are used to determine the variant of the simplex algorithm to solve the current restricted linear program. While the number of negative reduced cost variables added in an iteration, say t, is large, the dual simplex method is used, but if after a fixed number of iterations ($t > K_2$) the number of negative reduced cost variables added in an iteration is small ($|\mathcal{C}_t| < K_1$), we switch to using primal simplex method. The primal simplex method is more effective if only a few negative reduced costs have been added. Parameter N_{LP} is a limit on the total number of variables added. When N_{LP} needs to be set to infinity, the linear program is solved to optimality. However, when N_{LP} is set to a finite number, and the algorithm is terminated because this limit is reached, only an approximate solution to the linear program is obtained. Solving the linear program approximately can be considered in case solution times become prohibitive.

Algorithm 11: EFFICIENT-ENHANCED-BASIC(N_{iter},N_e,N_f,N_a,$Sort$,$Best$,ℓ,m)

$\mathcal{F}_1, \mathcal{E}_1 \leftarrow$ *unordered lists of facilities in the network and equipment categories*
$\mathcal{C} \leftarrow \{\}$
if $Sort$ **then**
  $\mathcal{E}_1 \leftarrow$ *Equipment categories sorted by current objective cost in non-increasing order*
  $k_1 \leftarrow 1$
else
  $k_1 \leftarrow \ell + 1$ *// index of last equipment category searched in previous iteration*
for ***each*** *equipment type* $\boldsymbol{e} = k_1, .., |\mathcal{E}_1|$ *in* $\mathcal{E}_1$ **do**
  $\underline{\pi}_e, \bar{\pi}_e \leftarrow$ *minimum and maximum of dual variables* π *for each facility*
  $\mathcal{C}_e \leftarrow \{\},\ r \leftarrow 0,\ r_{prev} \leftarrow 0$
  if $Sort$ **then**
    $\mathcal{F}_1 \leftarrow$ *facilities sorted by* $\underline{\pi}_e$ *in non-decreasing order*
    $k_2 \leftarrow 1$
  else
    $k_2 \leftarrow m_e + 1$ *// index of last facility searched in previous iteration for* e
  for ***each*** *facility* $\boldsymbol{i} = k_2, .., |\mathcal{F}_1|$ *in* $\mathcal{F}_1$ **do**
    if $\underline{\pi}_{ei} = 0$ **then**
      continue
    $Inbound[i] \leftarrow$ *unordered list of facilities* j *with arc* (j, i)
    if $Sort$ **then**
      $Inbound[i] \leftarrow$ *facilities* j *with arc* (j, i) *sorted by* $D_{(ji)e} - \bar{\pi}_{ej}$ *in non-decreasing order*
    $\mathcal{C}_f \leftarrow \{\}$
    for ***each*** *facility* $\boldsymbol{j}$ *in* $Inbound[i]$ **do**
      if $Sort$ *and* $D_{(ji)e} + \underline{\pi}_{ei} - \bar{\pi}_{ej} > 0$ **then**
        break
      if $Sort = False$ *and* $D_{(ji)e} + \underline{\pi}_{ei} - \bar{\pi}_{ej} > 0$ **then**
        continue
      $\mathcal{C}_a \leftarrow \{\}$ *// list of at most* N_a *negative reduced cost timed arcs (sorted)*
      $\mathcal{T}(i) \leftarrow$ *set of time-points at facility* i *in the order of time*
      for ***each*** *time-point* $\boldsymbol{t}$ *in* $\mathcal{T}(i)$ **do**
        if $D_{(ji)e} + \pi_{iet} \geq 0$ **then**
          break
        $a \leftarrow ((j, t_j), (i, t))$ *// available empty repositioning arc*
        $r \leftarrow D_{(ji)e} + \pi_{iet} - \pi_{jet_j}$ *// reduced cost of arc* a
        if $r = r_{prev}$ **then**
          continue // skipping arcs with the same reduced cost as the last one found
        $r_{prev} \leftarrow r$
        if $r < 0$ **then**
          $\mathcal{C}_a \leftarrow a$
        if $Best = False$ *and* $|\mathcal{C}_a| \geq N_a$ **then**
          break
      $\mathcal{C}_f \leftarrow \mathcal{C}_f \cup \mathcal{C}_a$
      if $|\mathcal{C}_f| \geq N_f$ **then**
        break
    $\mathcal{C}_e \leftarrow \mathcal{C}_e \cup \mathcal{C}_f$
    if $|\mathcal{C}_e| \geq N_e$ **then**
      break
  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_e$
  if $|\mathcal{C}| \geq N_{iter}$ **then**
    break
return $\mathcal{C}$

Algorithm 12: LP-HEUR(N_{LP},K_1,K_2,N_{iter},N_e,N_f,N_a,$Sort$,$Best$)

$\mathcal{LP} \leftarrow$ *LP relaxation of* $\mathcal{IAM}$ *model with an initial set of empty repositioning variables*
$Terminate \leftarrow False$
$\mathcal{C}_t \leftarrow \emptyset$
$t \leftarrow 0$
$\ell \leftarrow 0$
$m \leftarrow$ **0** (*vector of size* $|\mathcal{E}|$)
while $Terminate = False$ **do**
  Solve $\mathcal{LP}$ and retrieve values of dual variables
  $\mathcal{C}_t \leftarrow$ PRICING-ALGORITHM(N_{iter},N_e,N_f,N_a,$Sort$,$Best$,ℓ,m)
  if $\mathcal{C}_t = \emptyset$ **then**
    break
  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_t$
  if $|\mathcal{C}| \geq N_{LP}$ **then**
    break
  if $|\mathcal{C}_t| < K_1$ & $t > K_2$ **then**
    Switch to Primal Simplex when solving $\mathcal{LP}$
  $t \leftarrow t + 1$
  Update $\mathcal{LP}$ with new columns in $\mathcal{C}_t$
return $\mathcal{C}$

Target inventory constraints. So far, we have ignored any target inventory constraints. Unfortunately, when target inventory constraints are included, a few things change. The monotonicity property of the dual values remains true, as Constraints (4.11) are unchanged, but the non-positive property of dual values may no longer be satisfied when we enforce maximum target inventory constraints. Let α^l_{ie} and α^u_{ie} denote the dual variables associated with the minimum and maximum target inventory constraints respectively. Constraints (4.12) become

$$\pi_{ieT} + \alpha^l_{ie} - \alpha^u_{ie} \leq 0 \qquad \forall (i,T) \in \mathcal{N},\, e \in \mathcal{E}. \tag{4.19}$$

When maximum target inventory constraints are not present, we have

$$\pi_{ieT} \leq -\alpha^l_{ie} \leq 0 \qquad \forall (i,T) \in \mathcal{N},\, e \in \mathcal{E}, \tag{4.20}$$

which, because α^l_{ie} is non-negative, ensures non-positive dual values. However, in the presence of

maximum target inventory constraints, non-positive dual values can no longer be guaranteed.

4.5.3 Solving the IP

When the substitution variables y_{lc} are fixed, say at values $\bar{y}_{lc}$, then $\mathcal{IAM}$ reduces to a number of minimum cost flow problems, one for each equipment type, with flow variables s_{iet} and u_{ae} as represented in Figure 4.3.

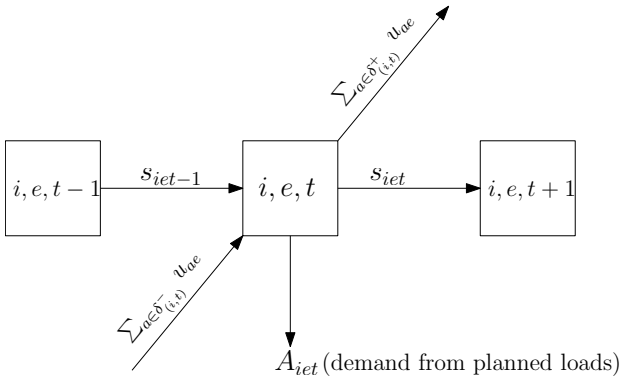


Figure 4.3: Inventory flow of equipment e at node (i, t).

Here A_{iet} represents the contribution of the planned loads to the inventory of equipment e at node (i, t); it can take on positive or negative values and is calculated as follows:

$$A_{iet} = \sum_{l \in \mathcal{L}^+_{(i,t)}} \sum_{c \in S_l} \eta_{ce} \bar{y}_{lc} - \sum_{l \in \mathcal{L}^-_{(i,t)}} \sum_{c \in S_l} \eta_{ce} \bar{y}_{lc}.$$

More specifically, the resulting problem is

$$\min \sum_{a\in\mathcal{A}} \sum_{e\in\mathcal{E}} D_a u_{ae} \tag{4.21}$$

$$\text{s.t.} \left(s_{iet_1} + \sum_{a\in\delta^+(i,t_1)} u_{ae} \right) - \left(\sum_{a\in\delta^-(i,t_1)} u_{ae} \right) = I_{ie} - A_{iet_1},\ \forall (i,t_1) \in \mathcal{N},\ e \in \mathcal{E}, \tag{4.22}$$

$$\left(s_{iet} + \sum_{a\in\delta^+(i,t)} u_{ae} \right) - \left(s_{iet^-} + \sum_{a\in\delta^-(i,t)} u_{ae} \right) = -A_{iet},\ \forall (i,t) \in \mathcal{N},\ t > 1,\ e \in \mathcal{E}, \tag{4.23}$$

$$s_{iet} \in \mathbb{Z}_{\geq 0}, \quad (i,t) \in \mathcal{N}, e \in \mathcal{E}, \tag{4.24}$$

$$u_{ae} \in \mathbb{Z}_{\geq 0}, \quad a \in \mathcal{A},\ e \in \mathcal{E}, \tag{4.25}$$

which, because there is no longer any interaction between equipment types, decomposes into $|\mathcal{E}|$ minimum cost flow problems. This suggests that a branching scheme that focuses on the substitution variables is appropriate for solving $\mathcal{IAM}$.

However, given that even solving the LP relaxation is time consuming for the size of instances that we are interested in, we employ, price-and-branch, a well-known heuristic scheme. In a price-and-branch scheme, the LP relaxation at the root node of the search tree is solved using dynamic pricing of variables, and after that an IP is solved using only the (partial) set of variables generated at the root node. This is a heuristic, because to obtain a proven optimal solution it will be necessary to dynamically generate variables at every node in the search tree (as in branch-and-price algorithms). Algorithm 13 gives the pseudo-code of IP-HEUR.

4.6 Computational Study

We have conducted a set of computational experiments to demonstrate the value of the proposed inventory-aware equipment management model ($\mathcal{IAM}$) for a package express carrier operating a

Algorithm 13: IP-HEUR(N_{LP},K_1,K_2,N_{iter},N_e,N_f,N_a,$Sort$,$Best$)

$\mathcal{IP} \leftarrow \mathcal{IAM}$ *model with initial set of empty repositioning variables*
$\mathcal{C} \leftarrow \emptyset$
$\mathcal{C} \leftarrow$ LP-HEUR(N_{LP},K_1,K_2,N_{iter},N_e,N_f,N_a,$Sort$,$Best$)
$\mathcal{IP} \leftarrow \mathcal{IAM}$ *model with expanded set of empty repositioning variables, i.e., including variables in* $\mathcal{C}$
Solve $\mathcal{IP}$

large ground service network with a large and heterogeneous fleet of trailers and containers. The experiments assess the computational efficiency of the proposed solution methodology and extract business insight regarding equipment management, by answering the following questions:

- What performance enhancements are achieved by more sophisticated variable pricing schemes, i.e., what performance improvements are observed when employing ENHANCED-BASIC and EFFICIENT-ENHANCED-BASIC rather than the naive pricing scheme BASIC?
- What is the impact of the pricing scheme parameters on the efficiency of solving the LP relaxation of $\mathcal{IAM}$ for a given time discretization?
- What is the impact of having a finer time discretization? What is the trade-off between efficiency (reducing the run-time) and quality (reducing the transportation cost)?
- What is the trade-off between leveraging equipment substitutions and introducing empty repositioning movements to ensure no equipment stock-outs during the planning period?
- What is the trade-off between the equipment fleet size and the empty repositioning costs?
- What is the impact of allowing substitutions involving composite equipment configurations?
- What is the impact of solving the final IP model heuristically?

4.6.1 Instances

We use a set of ten instances in the computational study. The instances are derived from historical weekly load planning data provided by a major U.S. package express carrier. The carrier's ground network has about 2300 facilities, which include company terminals, customer locations, and other locations where equipment inventory is monitored, e.g., rail yards. Table 4.1 summarizes relevant characteristics of the instances.

Table 4.1: Instance characteristics. A facility is considered active when there is at least one inbound or outbound load at the facility during the week. The fleet size is based on the equipment at an active facility and on the en-route equipment at the start of the planning period. The number of time-points is based on parameters $\tau_m = 30$ minutes and $\tau_M = 1$ day.

Instance	# Active Facilities	# Loads	Total Mies	# Time Points	Fleet Size	Empty Legs (%)	Bobtail Legs (%)
1	1,152	181,165	34,145,280	28,274	19,491	33.10	13.70
2	1,149	180,375	33,948,206	28,143	19,554	33.28	13.80
3	1,149	179,619	33,840,054	28,080	19,375	33.09	13.64
4	1,148	179,527	33,858,084	28,029	19,438	32.85	13.49
5	1,147	180,834	34,286,951	28,093	19,763	32.23	13.36
6	1,149	182,867	34,841,019	28,167	20,238	31.77	13.08
7	1,151	185,385	35,699,631	28,364	20,731	31.30	12.80
8	1,149	189,136	36,531,351	28,681	20,939	31.06	12.86
9	1,149	188,987	36,788,322	28,664	20,626	31.00	12.47
10	1,149	191,092	37,636,547	28,803	21,219	30.82	12.36

The similarities between the instances are a consequence of the fact that they are derived from consecutive weeks of data. Each instance is made up of a weekly load plan that contains all the loads that are scheduled to depart during the week. The timed loads are of two types: (a) loaded movements with an assigned equipment type and a specified volume (as a percentage of equipment capacity), and (b) empty movements with an assigned equipment type but without a specified volume. In addition to the timed loads, a load plan also contains a set of timed bobtail movements (needed to ensure driver cycles). All these movements have a fixed dispatch and arrival time. These times account for the time required for loading and unloading, so that the dispatch time

corresponds to the time equipment is taken from the yard and the arrival time corresponds to the time equipment is delivered to the yard. The instances have about 200 thousand movements, with about 55% of these being loaded, 32% being empty, and 13% being bobtails.

The carrier operates a heterogeneous fleet of 13 equipment types. These differ in terms of characteristics such as size (e.g., 53 foot trailers and 28 foot pups), intermodal compatibility (e.g. containers and trailers on flatcar), ownership (e.g., company, customer, or third party owned), etc. Table 4.2 gives the composition of the fleet for Instance 1. Only one composite configuration is

Table 4.2: Types and number of units of equipment for Instance 1

Equipment Id	# Units	Percentage (%)
1	77	0.40
2	748	3.84
3	16	0.08
4	3	0.02
5	21	0.11
6	12,541	64.34
7	609	3.12
8	183	0.94
9	73	0.37
10	294	1.51
11	487	2.50
12	542	2.78
13	3,897	19.99

allowed in the network, namely, the 2-pup train widely used in U.S. ground transportation. Each instance comes with an equipment allowance table that specifies for each load, a set of configurations of equipment types that can be assigned to the load. This table is used to generate the sets S_l for each load l.

Each instance includes a snapshot of the system at the start of the planning period. This snapshot includes the inventory of each facility-equipment type pair, and in-transit equipment that is expected to arrive at a facility during the planning period. As this information was not provided by the package express company, we artificially generated the initial inventories by using the load plan

as follows. For each facility-equipment type pair, we calculate, based on inbound and outbound loads, the minimum initial inventory required to ensure that there will be no equipment stock-out during the planning period. We then randomly choose an initial inventory level from a uniform distribution centered around the minimum required inventory. By doing so, each facility in the network has either an surplus or a deficit. A deficit implies that the facility will experience one or more shortages during the planning period unless equipment substitutions and empty repositioning moves are planned.

To account for the possibility of equipment stock-outs, we add an artificial equipment "source" at each node of the time-expanded expanded network and this source can be used to ensure that no stock-out occurs; a high-penalty is incurred when using an artificial source to discourage their use (we prefer the use of equipment substitutions and empty repositioning). The penalty for using an artificial source is set to the cost of movement of 4,649 miles (the longest distance between two locations in the network). All instances are such that if empty repositioning movements can be introduced at any time during the planning period, then stock-outs can be avoided by equipment substitutions and empty repositioning.

The inventory-aware model is coded in . Mixed integer programs are solved using the commercial solver Gurobi 9.0 with default settings. All experiments were run in a 20-core machine with Intel(R) Xeon(R) 2.30GHz processors and 256GB of RAM. The optimality tolerance is set to 0.005. No time limit was enforced.

4.6.2 Inventory-aware equipment management

We start by solving the instances with the EFFICIENT-ENHANCED-BASIC scheme, where we solve the LP relaxation to optimality ($N_{LP} = \infty$). Table 4.3 summarizes the results. We report the following statistics:

- **IP_OBJ**: objective value of the IP, i.e., the total miles of empty repositioning introduced,

- **LP_OBJ**: objective value of the LP relaxation,
- **# SUB**: number of loads for which the initial equipment type is replaced,
- **# ITER**: number of iterations, where the first iteration represents the solution of the LP relaxation without any empty repositioning variables,
- **# VAR**: total number of variables added,
- **VG_T**: total time spent searching negative reduced cost variables (in seconds),
- **LP_T**: total time spent solving LP relaxations (in seconds),
- **IP_T**: time spent solving the IP (in seconds),
- **T_T**: total time (in seconds).

Table 4.3: Results using IP-HEUR with default parameters $N_{LP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 6$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$.

Ins.	IP_OBJ	LP_OBJ	#SUB	#ITER	#VAR	VG_T	LP_T	IP_T	T_T
1	26,753	26,604	39,025	23	330,121	1,118	3,817	13,797	18,732
2	26,876	26,635	38,756	19	309,834	680	3,651	23,124	27,455
3	20,926	20,875	38,270	22	332,775	767	2,472	13,525	16,764
4	21,941	21,836	38,066	22	321,702	625	2,750	14,255	17,630
5	26,071	25,910	38,272	18	262,748	431	1,867	5,881	8,178
6	26,731	26,560	38,696	26	344,667	798	2,492	6,443	9,733
7	24,988	24,916	37,964	19	277,291	414	1,835	6,211	8,460
8	19,878	19,777	40,250	21	335,081	604	2,381	12,626	15,611
9	23,097	22,986	39,453	20	290,226	401	2,364	13,152	15,916
10	22,146	21,983	40,588	20	285,339	455	2,343	13,061	15,859

We observe that the difference between the objective value of the IP and the objective value of the LP relaxation is very small (less than 0.54% in final gap on average). This shows that our price-and-branch heuristic (Algorithm 13) is effective and little can be gained from a full-blown branch-and-price implementation. The LP and IP objective values represent the total empty

repositioning miles added to the original load plan. Comparing these values to the total miles in the original load plan (Table 4.1), we see that the increase is very small, less than 0.1%. In addition to new empty repositioning movements, the equipment configuration assigned to loads has been changed for about 40,000 loads (about 20% of the total number of loads) in the adjusted load plan.

We observe too that on average about 310,000 variables are generated during the solution of the LP relaxation and that on average this requires about 21 iterations. The total solution time is, on average, a bit less than 4 hours, of which about 4% is spent identifying negative reduced cost variables, about 18% is spent solving LPs, and about 78% of time is spent solving the IP. A total time of less than 4 hours is acceptable for the intended use of $\mathcal{IAM}$.

Next, we explore the trade-off between equipment substitution and empty repositioning decisions. To do so, we add constraint

$$\sum_{l \in \mathcal{L}} \sum_{\substack{c \in S_l \\ c \neq q(l)}} y_{lc} \leq Cap \tag{4.26}$$

to $\mathcal{IAM}$, which limits the number of substitutions, and we vary the right hand side. More specifically, we solve the LP allowing no subsitutions and then solve different IPs (with the variables of the final LP) for different limits on the number of substitutions (i.e., different values of Cap).

The results for nine different limits can be found in Table 4.4. The results clearly demonstrate the benefit of equipment substitutions when ensuring no equipment stock-outs as they decrease the repositioning costs by more than 65% on average.

For Instance 2, we show the trade-off curve in Figure 4.4. For this case, we need 77,771 repositioning miles to avoid equipment stock-outs when no equipment substitutions are allowed (i.e., $Cap = 0$) as opposed to only 27,668 when no limit is imposed on the number of substitutions (i.e., $Cap = \infty$).

Next, we explore the minimum number of equipment substitutions required to reach the minimum required repositioning miles. This is valuable in practice, because even though equipment

Table 4.4: Trade-off between empty repositioning and equipment substitutions (using IP-HEUR with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 6$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$.

Cap	0	50	100	200	500	1000	1,500	2,000	∞
Ins. 1	80,754	67,169	58,996	48,994	34,669	27,755	27,633	27,633	27,633
Ins. 2	77,771	65,208	58,406	49,449	35,683	27,840	27,668	27,668	27,668
Ins. 3	71,835	58,292	51,119	42,218	29,225	21,544	21,376	21,376	21,376
Ins. 4	71,120	59,330	52,550	43,679	30,269	22,623	22,412	22,412	22,412
Ins. 5	78,392	65,688	58,218	48,486	34,315	26,758	26,666	26,666	26,666
Ins. 6	80,764	67,682	59,843	49,707	35,052	27,451	27,255	27,255	27,255
Ins. 7	73,786	60,360	52,942	43,562	30,839	25,212	25,202	25,202	25,202
Ins. 8	72,938	58,729	50,958	41,358	27,323	20,698	20,668	20,667	20,667
Ins. 9	85,958	71,723	62,381	51,030	34,484	24,903	23,984	23,984	23,984
Ins. 10	70,397	59,082	51,641	41,986	28,719	22,910	22,906	22,905	22,905

substitutions are "free", planners like to adjust the initial load plan as little as possible (i.e., with the fewest equipment substitutions). For that, we take a hierarchical approach where we solve $\mathcal{IAM}$ in the first stage and minimize the number of equipment substitutions in the second stage forcing that the minimum repositioning costs found in the first stage do not change. The objective function of the second stage can be formulated as

$$\min \sum_{l \in \mathcal{L}} \sum_{\substack{c \in S_l \\ c \neq q(l)}} y_{lc} \tag{4.27}$$

and forcing that the minimum repositioning costs found in the first stage do not change is achieved by adding constraint

$$\sum_{a \in \mathcal{A}} \sum_{e \in \mathcal{E}} D_{ae} u_{ae} \leq \Omega^{\star} \tag{4.28}$$

where $\Omega^{\star}$ is the objective value of $\mathcal{IAM}$ model. For the ten instances, we find that this hierarchical approach results in a number of substitutions that is, on average, less than 1% of the total number

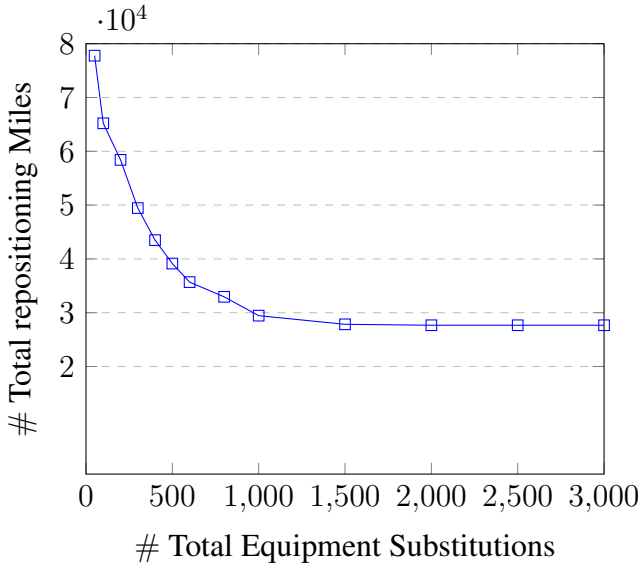


Figure 4.4: Relationship between the total repositioning cost required (in miles) and the limit on the number of substitutions allowed for Instance 2

of loads.

Finally, we explore the trade-off between the fleet size and the required empty repositioning cost. To do so, we vary the initial inventory of equipment in the network. More specifically, for each facility i and equipment type e, we adjust the inventory I_{ie} by multiplying it by a factor $\eta \geq 1$, i.e.,

$$\hat{I}_{ie} = \eta I_{ie},$$

where $\hat{I}_{ie}$ denotes the adjusted initial equipment inventory. The results can be found in Table 4.5, where **FS** represents the fleet size (with adjusted initial equipment inventory at the facilities. We observe that increasing the fleet size by 10% reduces the empty repositioning costs by about 30%. As less empty repositioning is required, we see that fewer empty repositioning variables have to be generated (about 15%), which requires fewer iterations (about 27%) and less time (about 29%).

Table 4.5: Impact of fleet size in IP-HEUR (with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 5$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$).

INS.	η	FS	IP-OBJ	LP-OBJ	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	1	19,491	26,753	26,604	328,529	17	397	1,729	5,971	8,096
	1.05	20,300	24,958	24,804	327,981	18	542	1,724	6,240	8,506
	1.10	21,371	18,156	17,975	279,623	12	359	1,440	5,604	7,403
2	1	19,554	26,876	26,635	283,467	10	168	1,407	14,820	16,394
	1.05	20,354	24,439	24,221	250,899	8	145	1,386	5,744	7,274
	1.10	21,458	17,872	17,753	262,690	9	200	1,192	4,502	5,894
3	1	19,375	20,926	20,875	331,735	17	397	1,738	9,691	11,826
	1.05	20,170	18,831	18,756	228,401	8	149	1,221	4,167	5,537
	1.10	21,267	15,611	15,604	239,553	9	182	1,309	2,898	4,389
4	1	19,438	21,941	21,836	321,614	19	409	2,103	5,156	7,668
	1.05	20,245	20,062	19,939	314,385	17	413	2,008	4,913	7,334
	1.10	21,307	16,486	16,399	239,219	10	218	1,536	5,339	7,093
5	1	19,763	26,071	25,910	222,557	9	146	1,305	3,557	5,007
	1.05	20,586	23,565	23,452	207,000	8	138	1,166	5,144	6,448
	1.10	21,690	16,650	16,603	208,589	9	172	1,323	3,528	5,022

4.6.3 Impact of Algorithmic Features and Choices

The performance of the price-and-branch heuristic, both in terms of the quality of the solution obtained and the efficiency with which this solution was produced, are impacted by many algorithmic features and choices. In this section, we assess this impact systematically.

Impact of the Discretization Scheme

To assess the impact of the discretization scheme on solution quality and algorithm efficiency, we conduct two experiments. First, we fix the minimum time between two consecutive time points, τ_m, to be 30 minutes and vary the maximum time between two consecutive time points, τ_M. Second, we fix the the maximum time between two consecutive time points, τ_M, to be 24 hours and vary the minimum time between two consecutive time points, τ_m. The goal is to quantify the impact of the number of time points as well as the organization of time points on quality and efficiency.

The results can be found in Tables 4.6 and 4.7, where **#TPT** represents the number of time-points, **OBJ** the repositioning cost, and **#P** the number of stock-outs (i.e., total number of equipment shortages observed at the time points). **VG-T**, **LP-T**, **IP-T**, and **TT** represent respectively the time (in seconds) spent in dynamic variable generation, solving the LP, solving the final IP, and the total run-time.

Table 4.6: Value of maximum time-step τ_M in the discretization and its impact on the performance of IP-HEUR (with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 6$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$).

INS.	τ_M	#TPT	OBJ	#P	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	168	26,110	26,482	1	278,450	25	1,311	4,298	21,070	26,679
	24	28,274	26,753	0	330,121	23	1,118	3,817	13,797	18,732
	12	34,503	26,593	0	438,856	31	1,114	4,413	9,029	14,556
2	168	25,993	26,050	1	282,064	20	819	3,901	20,048	24,767
	24	28,143	26,876	0	309,834	19	680	3,651	23,124	27,455
	12	34,357	26,700	0	433,501	23	597	3,165	6,728	10,490
3	168	25,930	22,873	0	286,696	22	1,181	4,024	24,448	29,652
	24	28,080	20,926	0	332,775	22	767	2,472	13,525	16,764
	12	34,307	20,669	0	426,588	26	788	3,696	9,367	13,851
4	168	25,876	24,081	0	251,744	20	791	3,050	14,487	18,328
	24	28,029	21,941	0	321,702	22	625	2,750	14,255	17,630
	12	34,264	21,886	0	419,037	29	1,616	6,465	13,908	21,989
5	168	25,949	25,549	1	220,819	17	677	2,842	13,427	16,946
	24	28,093	26,071	0	262,748	18	431	1,867	5,881	8,178
	12	34,313	25,655	0	408,998	29	1,959	6,275	12,420	20,655

The result in Table 4.6 show that reducing the maximum time between two consecutive time points from 168 to 12 hours eliminates stock-outs (Instances 1, 2, and 5) and reduces empty repositioning miles (Instances 3 and 4) as the number of repositioning options has increased. Even though the number of empty repositioning variables generated increases by about 62%, this does not always imply an increase in total time, as a larger number of variables typically implies shorter IP solve time. We also observe that the difference in repositioning costs between using $\tau_M = 24$ and $\tau_M = 12$ is small, less than 1%, but that using $\tau_M = 12$ appears to be more efficient (although results differ on different instances). The results clearly suggest that there is no need to reduce the

maximum time between consecutive time points even further.

Table 4.7: Value of minimum time-step τ_m in the discretization and its impact on the performance of IP-HEUR (with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 6$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$).

INS.	τ_m	#TPT	OBJ	#P	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	0	45,844	26,910	0	377,422	42	4,878	5,514	30,791	41,183
	0.5	28,274	26,753	0	330,121	23	1,118	3,817	13,797	18,732
	1	22,578	26,112	0	275,124	19	791	3,364	8,405	12,561
	2	17,544	25,508	0	208,341	15	388	2,225	8,333	10,946
2	0	45,593	26,931	0	390,788	25	1,845	5,466	27,815	35,126
	0.5	28,143	26,876	0	309,834	19	680	3,651	23,124	27,455
	1	22,525	26,478	0	296,670	20	694	3,439	9,555	13,687
	2	17,506	25,525	0	231,463	14	247	2,174	9,546	11,967
3	0	45,396	20,926	0	464,063	32	4,019	6,634	51,687	62,341
	0.5	28,080	20,926	0	332,775	22	767	2,472	13,525	16,764
	1	22,489	20,114	0	309,126	23	872	3,633	9,417	13,922
	2	17,495	19,917	0	224,790	15	363	2,239	7,049	9,651
4	0	45,333	22,004	0	389,565	29	3,304	7,252	24,596	35,151
	0.5	28,029	21,941	0	321,702	22	625	2,750	14,255	17,630
	1	22,436	21,297	0	259,833	20	624	3,316	12,187	16,127
	2	17,460	20,358	0	167,637	13	171	1,755	5,142	7,067
5	0	45,494	26,334	0	309,785	22	1,797	5,814	23,099	30,710
	0.5	28,093	26,071	0	262,748	18	431	1,867	5,881	8,178
	1	22,491	25,620	0	233,298	19	801	3,148	8,781	12,731
	2	17,452	25,056	0	162,416	14	268	2,465	5,778	8,511

The results in Table 4.7 show that enforcing a minimum time of one hour between two consecutive time points (i.e., only enforcing that inventory is monitored at least once every hour) greatly reduces the number of iterations (by about 30%) and the number of empty repositioning variables generated (by about 29%). This results in a reduction of total time of about 64%. It also reduces the empty repositioning costs (by about 3%), which is likely due to missing a few short periods of stock-outs (less than one hour). Given that in practice the variability in load and unload times is high (in the order of a few hours), it is reasonable to monitor the inventory using at least once an hour rather than more frequently.

Impact of enhanced variable generation schemes

As solving the LP relaxation represents a significant fraction of the total solution time, we have carefully designed the variable generation scheme. To evaluate the impact of the various ideas and techniques embedded in the variable generation schemes, we compare the efficiency of the three variable generation schemes as well as their impact on the quality of final IP solution (as the different schemes result in different sets of variables, the IP solutions may differ – as may the IP solution times). For ease of notation, we use B, E-B, and E-E-B to represent the BASIC, ENHANCED-BASIC, and EFFICIENT-ENHANCED-BASIC schemes respectively.

We incorporate one more technique to reduce the computation time of ENHANCED-BASIC: we terminate dynamic variable generation when the objective value has not changed for three consecutive iterations. In that case, it is likely that we have found the optimal LP objective value, but have not yet been able to prove it. This technique was already used by [48] to deal with the tailing-off behavior of column generation schemes. Another option would be to compute a lower bound on the objective value, as suggested in [49], and terminate when the optimality gap drops below a threshold. However, in our setting Farley's bound is weak and only produces tight lower bounds in the last few iterations. Therefore, we opted for the simple cut-off rule.

We also include a variation of ENHANCED-BASIC, which we refer to as ENHANCED-BASIC-RELAXED (E-B-R), in which we start with ENHANCED-BASIC, but switch to BASIC once the number of variables generated in an iteration drops below a threshold (20,000 in our experiments). The rationale behind this idea is that once only a relatively small number of variables is generated, diversity becomes less important and we no longer want to limit the search for negative reduced cost variables.

A summary of the results can be found in Table 4.8. The results clearly demonstrate the value of exploiting dual information as the EFFICIENT-ENHANCED-BASIC scheme is far more efficient than the BASIC and ENHANCED-BASIC schemes. More specifically, we see that the use of the

Table 4.8: Comparison of embedding the different variable generation schemes in IP-HEUR (with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 5$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$).

INS.	SCHEME	IP-OBJ	LP-OBJ	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	B	26,759	26,604	486,192	22	61,331	2,107	6,212	69,650
	E-B	26,759	26,604	332,224	19	46,257	1,972	4,985	53,214
	E-B-R	26,759	26,604	379,114	13	34,829	2,319	8,203	45,351
	E-E-B	26,753	26,604	328,529	17	397	1,729	5,971	8,096
2	B	26,875	26,635	286,526	17	74,203	2,457	12,240	88,900
	E-B	26,876	26,635	370,579	15	38,828	1,815	7,893	48,535
	E-B-R	26,876	26,635	401,141	13	29,816	1,484	4,393	35,693
	E-E-B	26,876	26,635	283,467	10	168	1,407	14,820	16,394
3	B	20,926	20,875	478,398	28	85,886	2,283	9,086	97,255
	E-B	20,962	20,875	296,908	12	65,227	2,953	24,824	93,004
	E-B-R	20,962	20,875	335,564	12	29,322	1,284	5,744	36,350
	E-E-B	20,926	20,875	331,735	17	397	1,738	9,691	11,826
4	B	21,941	21,836	439,791	20	59,004	2,054	8,801	69,859
	E-B	21,971	21,836	264,828	14	49,033	2,447	7,310	58,790
	E-B-R	21,971	21,836	320,145	11	26,062	1,194	4,842	32,098
	E-E-B	21,941	21,836	321,614	19	409	2,103	5,156	7,668
5	B	26,060	25,910	331,972	24	68,607	2,483	6,966	78,057
	E-B	26,071	25,910	215,569	9	24,606	2,039	12,423	39,068
	E-B-R	26,071	25,910	304,651	9	16,226	1,828	5,667	23,721
	E-E-B	26,071	25,910	222,557	9	146	1,305	3,557	5,007

EFFICIENT-ENHANCED-BASIC scheme reduces the total time by about 88% compared to BASIC and about 82% compared to ENHANCED-BASIC. The difference is even more pronounced when we compare the time spent in variable generation as the EFFICIENT-ENHANCED-BASIC scheme reduces this time by about 99.5% compared to BASIC and 99.3% compared to ENHANCED-BASIC. Importantly, the IP objective values reached by the different schemes are similar (the maximum difference is less than 0.1% for all instances).

Ensuring diversification in the initial iterations (ENHANCED-BASIC-RELAXED) pays off and achieves the smallest number of iterations. As expected, for most instances the BASIC scheme generated the largest number of variables.

In Figure 4.5, we present more detailed information about the solution process for Instance 5. We show for BASIC, ENHANCED-BASIC-RELAXED and EFFICIENT-ENHANCED-BASIC the objective value and the number of variables generated at each iteration. The effectiveness of the EFFICIENT-ENHANCED-BASIC scheme jumps out. The time per iteration is small and convergence to the optimal LP objective value is quick. It also generates fewer variables. (Note that we use a logarithmic scale on the horizontal axis, which obscures the large differences.)

Sensitivity analyses of dynamic variable generation

The EFFICIENT-ENHANCED-BASIC variable generation scheme has many control parameters (mostly aimed at diversifying the set of variables generated). Here, we conduct a sensitivity analysis to better understand the effect of these parameters, where we focus on computation time and number of variables generated. As a baseline, we use the following configuration EFFICIENT-ENHANCED-BASIC(40000,5000,100,5,True,True,ℓ,m). To assess the impact of different control parameters we use the following additional statistics:

- **AVG-CO**: average number of variables generated per iteration,
- **AVG-VG**: average generation time per iteration (in seconds),

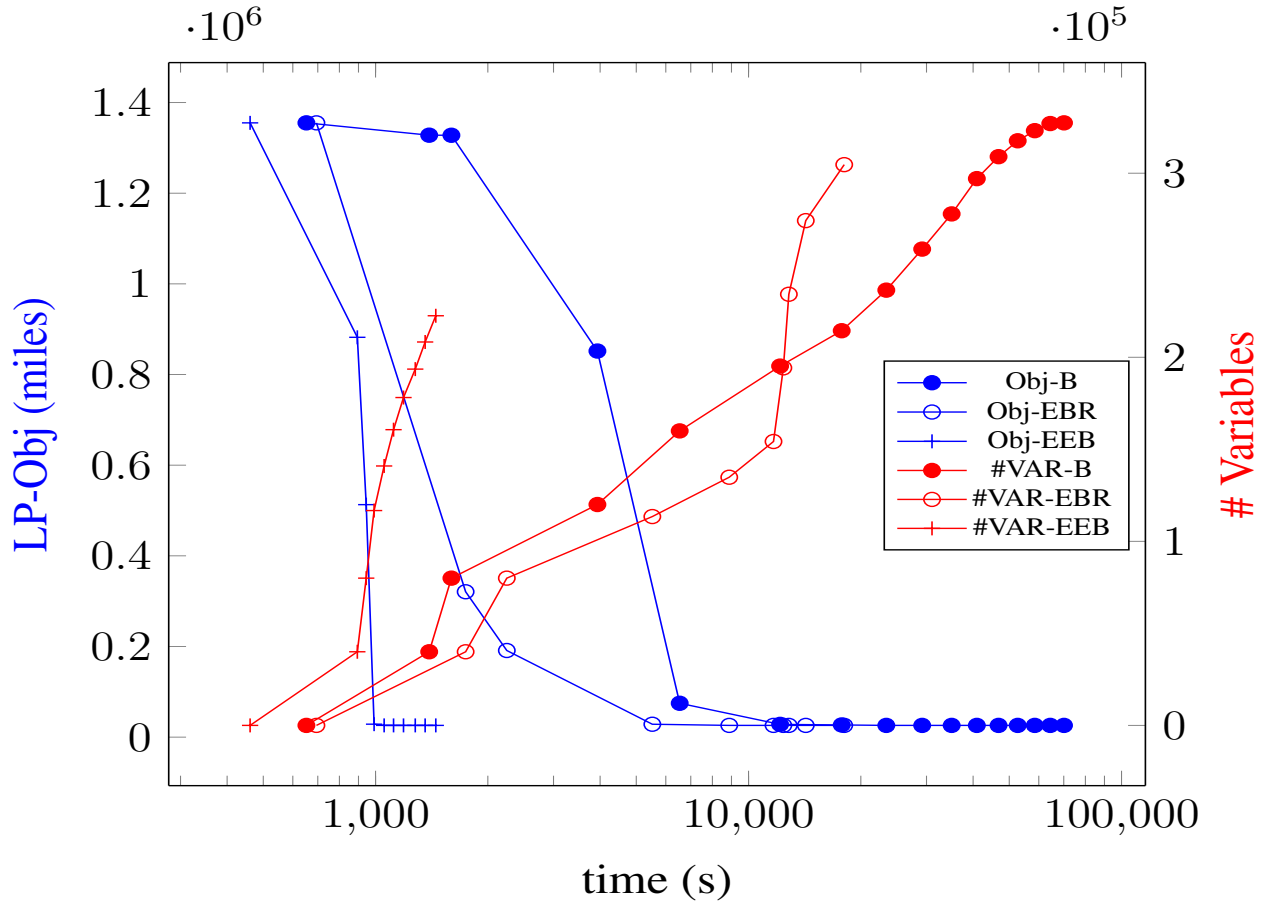


Figure 4.5: Comparison of the different variable generation schemes in terms of rate of convergence to the optimal objective value of the LP relaxation and the number of variables generated for Instance 5.

- **AVG-LP**: average LP solve time per iteration (in seconds),
- **AVG-OBj**: average change in objective function value per iteration (as a percentage),
- **AVG-R**: average ratio of the number of variables generated and the number of variables examined (i.e., including variables with non-negative reduced cost) per iteration (as a percentage),
- **T-T**: total LP solve time (in seconds).

Value of Sorting We solve each instance with sorting enabled and sorting disabled. When sorting

is disabled, a round robin scheme is used, as explained in Section 4.5.2, which also ensures some diversification. The results can be found in Table 4.9. We observe than when sorting is enabled,

Table 4.9: Impact of sorting on the performance of the EFFICIENT ENHANCED BASIC scheme.

INS.	Sort	#ITER	#VAR	AVG-CO	AVG-VG	AVG-LP	AVG-OBJ	AVG-R	T-T
1	True	23	330,121	14,353	44.15	119.60	8.50	5.58	4,513
	False	23	354,799	15,426	32.74	93.25	8.66	6.59	3,381
2	True	20	317,790	15,890	18.15	63.50	10.26	6.17	2,271
	False	25	381,998	15,280	36.94	81.49	8.68	6.22	3,791
3	True	22	332,775	15,126	43.62	114.20	9.51	6.25	4,506
	False	23	342,896	14,909	31.56	80.33	9.36	6.56	3,136
4	True	22	321,702	14,623	23.99	92.93	9.30	6.58	3,064
	False	22	333,186	15,145	20.52	81.82	9.89	7.37	2,692
5	True	18	262,748	14,597	26.33	93.90	10.62	6.67	2,641
	False	19	263,277	13,857	10.86	79.15	11.61	8.07	2,091

we generate fewer variables (about 6%) and take less time (about 15%).

Value of diversity We assess the value of the diversity created by limiting the number of variables generated for a single facility and a single arc, i.e., N_f and N_a. We compare combinations $(50, 3)$, $(100, 6)$, and $(200, 12)$. The results can be found in Table 4.10. We observe that when we relax enforcing diversity, i.e, $(N_f, N_a) = (200, 12)$, we generate more variables (about 57%) and increase solution time (about 52%) than when we favor diversity, i.e., $(N_f, N_a) = (50, 3)$.

Value of limits We assess the value of limiting the number of variables generated per iteration N_{iter} (so that new, hopefully more useful, dual information is obtained) and for an equipment type N_e (a high level mechanism to ensure diversity). We compare combinations $(8,000; 1,000)$, $(40,000; 5,000)$, $(80,000; 10,000)$, and $(120,000; 15,000)$. The results can be found in Table 4.11. We observe that generating too few variables per iteration has a negative effect on solution time (too many iterations), but so does generating too many variables per iteration (solving LP relaxations takes too long).

Table 4.10: Impact of diversity parameters N_f and N_a on the performance of the EFFICIENT ENHANCED BASIC scheme.

INS.	(N_f, N_a)	#ITER	#VAR	AVG-CO	AVG-VG	AVG-LP	AVG-OB	AVG-R	T-T
1	(50,3)	32	252,267	7,883	18.38	40.53	6.57	2.62	2,440
	(100,6)	23	330,121	14,353	44.15	119.60	8.50	5.58	4,513
	(200,12)	26	398,096	15,311	21.93	60.64	8.47	7.19	2,379
2	(50,3)	27	267,990	9,926	30.12	83.86	8.12	3.47	4,333
	(100,6)	19	309,834	16,307	33.84	126.04	10.30	6.92	4,297
	(200,12)	24	365,650	15,235	36.03	103.47	8.69	7.75	4,061
3	(50,3)	30	264,969	8,832	40.45	79.01	7.22	3.05	4,586
	(100,6)	22	332,775	15,126	43.62	114.20	9.51	6.25	4,506
	(200,12)	29	392,103	13,521	48.45	114.01	7.75	6.02	5,366
4	(50,3)	19	240,483	12,657	34.19	116.31	12.11	4.53	4,086
	(100,6)	22	321,702	14,623	23.99	92.93	9.30	6.58	3,064
	(200,12)	29	389,674	13,437	35.55	115.33	7.95	7.40	5,226
5	(50,3)	15	192,347	12,823	12.85	71.12	15.19	5.65	1,650
	(100,6)	18	262,748	14,597	26.33	93.90	10.62	6.67	2,641
	(200,12)	29	345,508	11,914	39.39	119.69	7.44	6.11	5,383

Value of Initialization Starting with an initial set of empty repositioning variables may result in more useful dual information early on in the solution process. Therefore, we compare starting without empty repositioning variables and starting with a set of initial empty repositioning variables (generated using Algorithm 14). The results can be found in Table 4.12. We observe that starting with an initial set of empty repositioning variables has few, if any, benefits; the solution time increases by about 5% (on average). In a real-life environment, where load plans do not change significantly from week to week, initializing with the set of empty repositioning movements performed in the preceding week may be beneficial.

Impact of composite configurations

Next, we investigate the value of allowing subsititutions involving composite configurations (when using EFFICIENT ENHANCED BASIC). We compare the baseline results to the case where we do not allow substitutions to a 2-pup train configuration, the only composite configuration in the

Table 4.11: Impact of limits N_{iter} and N_e on the performance of the EFFICIENT ENHANCED BASIC scheme.

INS.	(N_{iter}, N_e)	#ITER	#VAR	AVG-CO	AVG-VG	AVG-LP	AVG-OB	AVG-R	T-T
1	(8,000;1,000)	32	118,282	3,696	18.53	103.92	9.53	7.69	5,029
	(40,000;5,000)	24	338,543	14,106	24.88	72.97	8.15	5.22	2,837
	(80,000;10,000)	19	388,553	20,450	25.97	52.42	11.50	5.00	1,710
	(120,000;15,000)	18	473,239	26,291	36.41	75.71	9.75	4.54	2,321
2	(8,000;1,000)	33	144,352	4,374	14.45	53.32	9.35	7.06	3,709
	(40,000;5,000)	20	317,790	15,890	18.15	63.50	10.26	6.17	2,271
	(80,000;10,000)	19	408,557	21503	24.60	60.75	11.64	5.37	1,989
	(120,000;15,000)	17	470,995	27,706	30.97	82.70	10.41	4.92	2,402
3	(8,000;1,000)	28	115,072	4,110	7.35	25.68	10.87	8.20	1,294
	(40,000;5,000)	21	330,740	15,750	39.44	98.69	10.01	6.11	3,490
	(80,000;10,000)	21	416,621	19,839	25.31	55.14	10.60	4.77	1,893
	(120,000;15,000)	17	455,656	26,803	37.51	82.89	9.08	4.83	2,430
4	(8,000;1,000)	26	105,226	4,047	7.76	52.01	12.14	9.32	2,187
	(40,000;5,000)	25	303,779	12,151	18.78	78.76	8.85	5.73	2,898
	(80,000;10,000)	16	364,298	22,769	32.84	67.70	14.09	5.71	1,905
	(120,000;15,000)	16	441,432	27,590	34.75	95.21	10.19	5.04	2,458
5	(8,000;1,000)	19	92,184	4,852	6.63	55.45	16.75	12.89	1,772
	(40,000;5,000)	18	248,401	13,800	16.25	66.85	10.55	6.86	1,896
	(80,000;10,000)	16	334,042	20,878	17.58	56.83	13.96	5.96	1,423
	(120,000;15,000)	17	465,603	27,388	32.34	109.90	9.75	5.50	2,714

Algorithm 14: INITIALIZATION(N_{iter},N_e,N_f,N_a,$Sort$,ϵ)

$\mathcal{F}_1, \mathcal{E}_1 \leftarrow$ *unordered lists of facilities in the network and equipment categories*
$\mathcal{A}_1 \leftarrow \{\}$
if $Sort$ **then**
  $\mathcal{E}_1 \leftarrow$ *Equipment categories sorted by number of stock-outs in non-increasing order*
for ***each*** *equipment type* ***e*** *in* $\mathcal{E}_1$ **do**
  $\underline{I}_e \leftarrow$ *minimum of inventory level for each facility*
  $\mathcal{C}_e \leftarrow \{\}$
  if $Sort$ **then**
    $\mathcal{F}_1 \leftarrow$ *facilities sorted by* $\underline{I}_e$ *in non-decreasing order*
  for ***each*** *facility* ***i*** *in* $\mathcal{F}_1$ **do**
    if $\underline{I}_{ei} \geq 0 \, and \, Sort$ **then**
      break
    if $\underline{I}_{ei} \geq 0 \, and \, Sort = F$ **then**
      continue
    $Inbound[i] \leftarrow$ *unordered list of facilities* j *with arc* (j, i)
    if $Sort$ **then**
      $Inbound[i] \leftarrow$ *facilities* j *with arc* (j, i) *sorted by* D_{jie} *in non-decreasing order*
    $\mathcal{C}_f \leftarrow \{\}$
    $\mathcal{T}(i) \leftarrow$ *set of time-points at facility* i *in the order of time*
    for ***each*** *facility* ***j*** *in* $Inbound[i]$ **do**
      if $\underline{I}_{ej} < \epsilon * |\underline{I}_{ei}|$ **then**
        continue
      $\mathcal{C}_a \leftarrow \{\}$ *// list of at most* N_a *negative reduced cost timed arcs (sorted)*
      for ***each*** *time-point* ***t*** *in* $\mathcal{T}(i)$ **do**
        if $I_{iet} \geq 0$ **then**
          continue
        $a \leftarrow ((j, t_j), (i, t))$ *// available empty repositioning arc*
        $\mathcal{C}_a \leftarrow a$
        if $|\mathcal{C}_a| \geq N_a$ **then**
          break
      $\mathcal{C}_f \leftarrow \mathcal{C}_f \cup \mathcal{C}_a$
      if $|\mathcal{C}_f| \geq N_f$ **then**
        break
    $\mathcal{C}_e \leftarrow \mathcal{C}_e \cup \mathcal{C}_f$
    if $|\mathcal{C}_e| \geq N_e$ **then**
      break
  $\mathcal{A}_1 \leftarrow \mathcal{A}_1 \cup \mathcal{C}_e$
  if $|\mathcal{A}_1| \geq N$ **then**
    break
return $\mathcal{A}_1$

Table 4.12: Impact of initializing with the set of empty repositioning variables generated by Algorithm 14 with parameters $N_{iter} = 100,000$, $N_e = 10,000$, $N_f = 500$, $N_a = 10$, $Sort = True$, $\epsilon = 0.1$ on the performance of the EFFICIENT ENHANCED BASIC scheme.

INS.	$\mathcal{A}_1$	#ITER	#VAR	AVG-CO	AVG-VG	AVG-LP	AVG-OB	AVG-R	T-T
1	0	23	330,121	14,353	44.15	119.60	8.50	5.58	4,513
	938	24	334,320	13,930	53.55	113.46	9.17	4.83	4,960
2	0	20	317,790	15,890	18.15	63.50	10.26	6.17	2,271
	994	19	315,351	16,597	33.17	103.91	11.99	6.72	3,731
3	0	22	332,775	15,126	43.62	114.20	9.51	6.25	4,506
	1,713	22	324,951	14,771	49.04	100.93	9.05	5.39	4,193
4	0	22	321,702	14,623	23.99	92.93	9.30	6.58	3,064
	1,808	21	316,971	15,094	26.19	87.73	9.46	6.06	2,803
5	0	18	262,748	14,597	26.33	93.90	10.62	6.67	2,641
	2,164	21	270,507	12881	18.48	52.16	9.20	5.46	1,706

setting considered. The results can be found on Table 4.13.

Table 4.13: Impact of composite configurations on the repositioning cost and the performance of the EFFICIENT ENHANCED BASIC scheme.

INS.	2PUP ENABLED	IP-OBJ	LP-OBJ	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	True	26,753	26,604	328,529	17	397	1,729	5,971	8,096
	False	33,681	33,681	226,527	16	269	1,532	3,115	4,916
2	True	26,876	26,635	283,467	10	168	1,407	14,820	16,394
	False	36,485	36,485	246,280	11	110	994	1,657	2,762
3	True	20,926	20,875	331,735	17	397	1,738	9,691	11,826
	False	28,046	28,046	245,661	11	126	1,021	1,219	2,366
4	True	21,941	21,836	321,614	19	409	2,103	5,156	7,668
	False	29,586	29,586	231,141	11	137	1,165	1,929	3,231
5	True	26,071	25,910	222,557	9	146	1,305	3,557	5,007
	False	34,988	34,988	203,943	9	131	1,253	2,017	3,401

We observe that the repositioning costs increase by about 33% when we do not allow substitutions involving composite configurations. However, the overall run-time decreases by about 58% and by about 66% for the final IP model. This suggests that a substitution-based decomposition heuristic as used in Chapter 3 to solve the inventory-aware equipment management model might improve the run-time. We explore this in the next section.

We also observe that when we do not allow substitutions involving composite configurations the objective value of the LP relaxation always matches the objective value of the IP. This is due to the fact that in this case the model can be formulated as a multi-commodity network flow problem with side constraints on a directed acyclic graph (DAG) for which an optimal solution to the LP relaxation is often integral. This behavior is reported by many other researchers, i.e., an optimal solution to the LP-relaxation of an instance of a multi-commodity network flow problem often is integral even though the coefficient matrix is not totally unimodular, see, for example, [50], [51], [52], and [53].

Solving the IP model with a substitution-based decomposition heuristic

Here, we assess the benefits of solving the final IP model with SUB-HEUR, the substitution-based decomposition heuristic presented in Chapter 3. As solving the final IP model of takes about 80% of the total time of IP-HEUR using SUB-HEUR may result in a significant speed-up. We compare the performance of the two variants, i.e., solving the last IP model exactly, EXACT, and solving the last IP model heuristically, SUB-HEUR, using the repositioning cost and the run-time. The results can be found in Table 4.14.

Table 4.14: Performance of the substitution decomposition based heuristic SUB-HEUR in solving the final IP model.

INS.	IP-SCHEME	IP-OBJ	IP-T
1	EXACT	26,753	5,971
	SUB-HEUR	36,353	444
2	EXACT	26,876	14,820
	SUB-HEUR	27,027	703
3	EXACT	20,926	9,691
	SUB-HEUR	21,463	455
4	EXACT	21,941	5,156
	SUB-HEUR	22,423	550
5	EXACT	26,071	3,557
	SUB-HEUR	26,581	234

We observe that using SUB-HEUR to solve the final IP model reduces the IP solution time by about 93% on average – which implies reducing the total time by about 73% on average. On the other hand, the repositioning costs increase by about 9% on average. However, this percentage is skewed by a single outlier; Instance 1 saw an increase in costs of 36%, whereas the remaining instances saw an increase in costs of less than 2%.

4.6.4 Exact Methods

In the previous computational study, we have shown how to obtain a high-quality, but not necessarily optimal solution in a reasonable amount of time. In this section, we discuss two approaches that can be considered to obtain an proven optimal solution: branch and price and Benders decomposition.

Branch and Price

To solve the IP model to optimality, a branch and price algorithm (which combines branch and bound with column generation) can be implemented, which can leverage the methodology of Section 4.5.2. In a branch and price algorithm, the LP relaxation at every node in the search tree is solved using column generation. At a node, starting from a small subset of columns, the LP relaxation can be proven to be optimal by solving a pricing problem; the pricing searches for new columns that might improve the current solution. If such columns are found, they are added to the (restricted) LP, and the latter is resolved. When no such columns are found, the solution to LP relaxation is optimal. If the solution to the LP is not integral (and the node cannot be fathomed) branching occurs, and the process is repeated.

An important component of a branch and price algorithm the branching scheme. For our problem, we can use a previous result to define a branching scheme. We have shown that when substitution variables y_{lc} are fixed, the resulting model can be formulated and solved as a set of minimum cost flow problems (one for each equipment category). This observation suggests a branching

scheme that only considers substitution variables y_{lc} as variables to branch on. Given that the repositioning variables, which are generated dynamically, are independent from the substitution variables, the branching decisions do not affect the structure of the pricing problem. A branching decision of the form $y_{lc} = 0$ only impacts the set S_l of eligible configurations for load l as we have to remove c from this set. Similarly, a branching decision of the form $y_{lc} = 1$ amounts to restricting S_l to configuration c for load l. All the repositioning arcs in the $\mathcal{A}$ remain eligible and can be generated and used at any point of the branch and price algorithm. In the branching strategy, a practical choice is to give priority to substitutions involving composite configurations (e.g., 2 pup trains) as we observed in extensive computational experimentation that the solution from the LP relaxation is typically integral when only one-to-one substitutions are allowed.

Benders Decomposition

As stated above, when the substitution variables y_{lc} are fixed, solving the problem boils down to solving $|\mathcal{E}|$ minimum-cost flow problems, one for each equipment category. This suggests that Benders decomposition ([54]) might be a suitable approach for solving the original MIP problem. The idea is to decompose the set of variables in the problem into two groups. The first group contains variables that make the problem difficult to solve, which in our case correspond to the substitution variables y_{lc}. These variables form a Relaxed Master Problem, $\mathcal{RMP}$, that can be solved iteratively to find an optimal solution of the original problem. The second group contains variables that make up sub-problems that are (usually) easier to solve, which in our case correspond to the inventory and empty repositioning variables, s_{iet} and u_{ae}. Solving the sub-problems enables to derive cuts that are added to $\mathcal{RMP}$. These cuts are referred to as *Benders cuts*.

We start by formulating the sub-problem $\mathcal{SP}_e$ for a given equipment category e in $\mathcal{E}$. We

assume the relaxed master problem has a solution $\bar{y}$.

$$\mathcal{SP}_e(\bar{y}): \quad \min \sum_{a\in\mathcal{A}} D_{ae}u_{ae} \tag{4.29}$$

$$\text{s.t.} \left(s_{ie1} + \sum_{a\in\delta^+_{(i,1)}} u_{ae}\right) - \left(\sum_{a\in\delta^-_{(i,1)}} u_{ae}\right) = I_{ie0} - A_{ie1}(\bar{y}),\ \forall (i,1)\in\mathcal{N}, \tag{4.30}$$

$$\left(s_{iet} + \sum_{a\in\delta^+_{(i,t)}} u_{ae}\right) - \left(s_{ie(t-1)} + \sum_{a\in\delta^-_{(i,t)}} u_{ae}\right) = -A_{iet}(\bar{y}),\ \forall (i,t)\in\mathcal{N},\ t>1, \tag{4.31}$$

$$s_{iet}\in\mathbb{Z}_{\geq 0}, \quad (i,t)\in\mathcal{N}, \tag{4.32}$$

$$u_{ae}\in\mathbb{Z}_{\geq 0}, \quad a\in\mathcal{A}, \tag{4.33}$$

where $A_{iet}(\bar{y}) = \sum_{l\in\mathcal{L}^+_{(i,t)}} \sum_{c\in S_l} \eta_{ce}\bar{y}_{lc} - \sum_{l\in\mathcal{L}^-(i,t)} \sum_{c\in S_l} \eta_{ce}\bar{y}_{lc}$.

We also formulate the dual of $\mathcal{SP}_e$, denoted by $\mathcal{DSP}_e$:

$$\mathcal{DSP}_e(\bar{y}): \quad \max \sum_{(i,0)\in\mathcal{N}} \alpha_{ie1}I_{ie0} - \sum_{(i,t)\in\mathcal{N}} \alpha_{iet}A_{iet}(\bar{y}) \tag{4.34}$$

$$\text{s.t. } \alpha_{iet} - \alpha_{ie(t+1)} \leq 0,\ \forall (i,t)\in\mathcal{N},\ t = 1..(n_T-1), \tag{4.35}$$

$$\alpha_{ien_T} \leq 0,\ \forall (i,n_T)\in\mathcal{N}, \tag{4.36}$$

$$\alpha_{iet} - \alpha_{jet'} \leq D_{ae},\ \forall a = ((i,t)\to(j,t'))\in\mathcal{A}, \tag{4.37}$$

$$\alpha_{iet} \text{ free} \quad \forall (i,t)\in\mathcal{N}, t\neq 0. \tag{4.38}$$

This dual problem can be expressed in terms of the extreme points α^j of the dual feasible space as

follows

$$\min q_e \tag{4.39}$$

$$\text{s.t. } q_e \geq \sum_{(i,0)\in\mathcal{N}} \alpha^j_{ie1} I_{ie0} - \sum_{(i,t)\in\mathcal{N}} \alpha^j_{iet} A_{iet}(\bar{y}),\ \forall j \in \mathcal{J}, \tag{4.40}$$

$$q_e \text{ free}, \tag{4.41}$$

where $\mathcal{J}$ is the set of indexes of the extreme points in the dual feasible space.

The original problem can now be formulated as follows:

$$\min \sum_{e\in\mathcal{E}} q_e \tag{4.42}$$

$$\text{s.t. } q_e \geq \sum_{(i,0)\in\mathcal{N}} \alpha^E_{ie1} I_{ie0} - \sum_{(i,t)\in\mathcal{N}} \alpha^j_{iet} A_{iet}(y),\ \forall e \in \mathcal{E},\ j \in \mathcal{J}, \tag{4.43}$$

$$\sum_{c\in S_l} y_{lc} = 1, \quad \forall l \in \mathcal{L}, \tag{4.44}$$

$$y_{lc} \in \{0,1\}, \quad \forall l \in \mathcal{L},\ c \in S_l, \tag{4.45}$$

$$q_e \text{ free}. \tag{4.46}$$

In this formulation, constraints (4.43) can yield a prohibitively large number of rows as all extreme points are considered. The value of Benders decomposition lies in the fact that it generates constraints (4.43) iteratively and parsimoniously starting from an empty set until it reaches optimality.

We define the relaxed master problem $\mathcal{RMP}$ as follows:

$$\mathcal{RMP} \qquad \min \sum_{e \in \mathcal{E}} q_e \tag{4.47}$$

$$\text{s.t. } q_e \geq \sum_{(i,0) \in \mathcal{N}} \alpha^j_{ie1} I_{ie0} - \sum_{(i,t) \in \mathcal{N}} \alpha^j_{iet} A_{iet}(y), \; \forall e \in \mathcal{E}, \; j \in \mathcal{B}, \tag{4.48}$$

$$\sum_{c \in S_l} y_{lc} = 1, \quad l \in \mathcal{L}, \tag{4.49}$$

$$y_{lc} \in \{0, 1\}, \quad \forall l \in \mathcal{L}, \; c \in S_l, \tag{4.50}$$

$$q_e \text{ free}, \forall e \in \mathcal{E}, \tag{4.51}$$

where $\mathcal{B}$ is the set of indexes of extreme points that will be generated by solving the sub-problems $\mathcal{SP}_e(\bar{y})$ at every iteration.

These ideas are combined in the Benders Decomposition algorithm for solving the problem presented below:

- **Step 0**: Initialize $\mathcal{B}$ to be the empty set,
- **Step 1**: Solve the Relaxed Master problem $\mathcal{RMP}$ to get $(\bar{y}, \bar{q})$,
- **Step 2**: Solve the sub-problems $\mathcal{SP}_e(\bar{y})$ (using EFFICIENT-ENHANCED-BASIC algorithm) and collect the solution $\bar{u}_{ae}$ and the dual solution $\bar{\alpha}^j$,
- **Step 3**: If $\sum_{e \in \mathcal{E}} \bar{q}_e = \sum_{e \in \mathcal{E}} \sum_{a \in \mathcal{A}} D_{ae} \bar{u}_{ae}$, go to **Step 5**,
- **Step 4**: Otherwise, add index of extreme point $\bar{\alpha}^j$ to $\mathcal{B}$. Return to **Step 1**,
- **Step 5**: The solution of the $\mathcal{RMP}$ is optimal. End.

In this algorithm, we use disaggregated Benders cuts and generate one cut per equipment type at each iteration. Alternatively, we can aggregate them and generate a single cut per iteration. To do

so, we define the variable $q = \sum_{e \in \mathcal{E}} q_e$ and we replace Constraints (4.48) by their sum over all equipment types:

$$q \geq \sum_{e \in \mathcal{E}} \sum_{(i,0) \in \mathcal{N}} \alpha^j_{ie1} I_{ie0} - \sum_{(i,t) \in \mathcal{N}} \alpha^j_{iet} A_{iet}(y),\ \forall j \in \mathcal{B}. \tag{4.52}$$

The advantage of this aggregation is the fact that we introduce fewer cuts. The disadvantage is that it results in a weaker formulation of the relaxed master problem.

Given that the sub-problems $\mathcal{SP}_e$ are network flow problems, they exhibit degeneracy which makes the convergence of Benders decomposition slow as a large number of Benders cuts is required. Magnanti and Wong ([55]) propose to use stronger Pareto optimal Benders cuts as a solution to accelerate the conversion. This method requires, at every iteration, to use an interior point of the relaxed master problem $\mathcal{RMP}$ that we denote $\hat{y}$ and solve an auxiliary sub-problem that we denote $\mathcal{SP}2_e$ after solving the sub-problems $\mathcal{SP}_e$. Let $\bar{\alpha}^j$ represent the dual solution of the problems $\mathcal{SP}_e$ The dual of the auxiliary sub-problem $\mathcal{SP}2_e$ can be formulated as follows:

$$\mathcal{DSP}2_e(\bar{y}, \hat{y}): \qquad \max \sum_{(i,0) \in \mathcal{N}} \alpha_{ie1} I_{ie0} - \sum_{(i,t) \in \mathcal{N}} \alpha_{iet} A_{iet}(\hat{y}) \tag{4.53}$$

$$\text{s.t. } \alpha_{iet} - \alpha_{ie(t+1)} \leq 0,\ \forall (i,t) \in \mathcal{N},\ t = 1..(n_T - 1), \tag{4.54}$$

$$\alpha_{ien_T} \leq 0,\ \forall (i, n_T) \in \mathcal{N}, \tag{4.55}$$

$$\alpha_{iet} - \alpha_{jet'} \leq D_{ae},\ \forall a = ((i,t) \rightarrow (j,t')) \in \mathcal{A}, \tag{4.56}$$

$$-\sum_{(i,0) \in \mathcal{N}} \alpha_{ie1} I_{ie0} + \sum_{(i,t) \in \mathcal{N}} \alpha_{iet} A_{iet}(\bar{y}) \leq -\sum_{(i,0) \in \mathcal{N}} \bar{\alpha}_{ie1} I_{ie0} + \sum_{(i,t) \in \mathcal{N}} \bar{\alpha}_{iet} A_{iet}(\bar{y}), \tag{4.57}$$

$$\alpha_{iet} \text{ free} \quad \forall (i,t) \in \mathcal{N}, t \neq 0, ,$$

$\mathcal{SP}2_e$ can be formulated as follows:

$$\mathcal{SP}2_e(\bar{y},\hat{y}): \quad \min \sum_{a\in\mathcal{A}} D_{ae}u_{ae} - \Big(\sum_{(i,0)\in\mathcal{N}} \bar{\alpha}_{ie1}I_{ie0} - \sum_{(i,t)\in\mathcal{N}} \bar{\alpha}_{iet}A_{iet}(\bar{y})\Big)\xi \tag{4.58}$$

$$\text{s.t.} \left(s_{ie1} + \sum_{a\in\delta^+_{(i,1)}} u_{ae}\right) - \left(\sum_{a\in\delta^-_{(i,1)}} u_{ae}\right) - (I_{ie0} - A_{ie1}(\bar{y}))\xi = I_{ie0} - A_{ie1}(\hat{y}),\ \forall (i,1)\in\mathcal{N}, \tag{4.59}$$

$$\left(s_{iet} + \sum_{a\in\delta^+_{(i,t)}} u_{ae}\right) - \left(s_{ie(t-1)} + \sum_{a\in\delta^-_{(i,t)}} u_{ae}\right) + A_{iet}(\bar{y})\xi = -A_{iet}(\hat{y}),\ \forall (i,t)\in\mathcal{N},\ t>1, \tag{4.60}$$

$$s_{iet}\in\mathbb{Z}_{\geq 0}, \quad (i,t)\in\mathcal{N}, \tag{4.61}$$

$$u_{ae}\in\mathbb{Z}_{\geq 0}, \quad a\in\mathcal{A}, \tag{4.62}$$

$$\xi\geq 0. \tag{4.63}$$

The dual solution of $\mathcal{SP}2_e$ gives a Pareto-Optimal Benders cut.

Performance of Benders decomposition

To evaluate the potential of Benders decomposition, we use smaller instances obtained by shortening the horizon of the original instances to two days and using a coarse discretization parameter $\tau_m = 2$ hours. We refer to theses instances as Instance 11-14.

As expected, using standard Benders cuts results in a slow rate of convergence as the sub-problems have a network flow structure. We refer to the Benders decomposition with standard cuts as BD-DISAGG and show an example of the rate of convergence in Figure 4.6 (using Instance 14 and showing the rate of convergence of both the relaxed master problem and the sub-problems). For this instance, the E-E-B heuristic finds an optimal solution in less than 25 seconds; Benders decomposition, after 5 hours, ends with a gap of 107%.

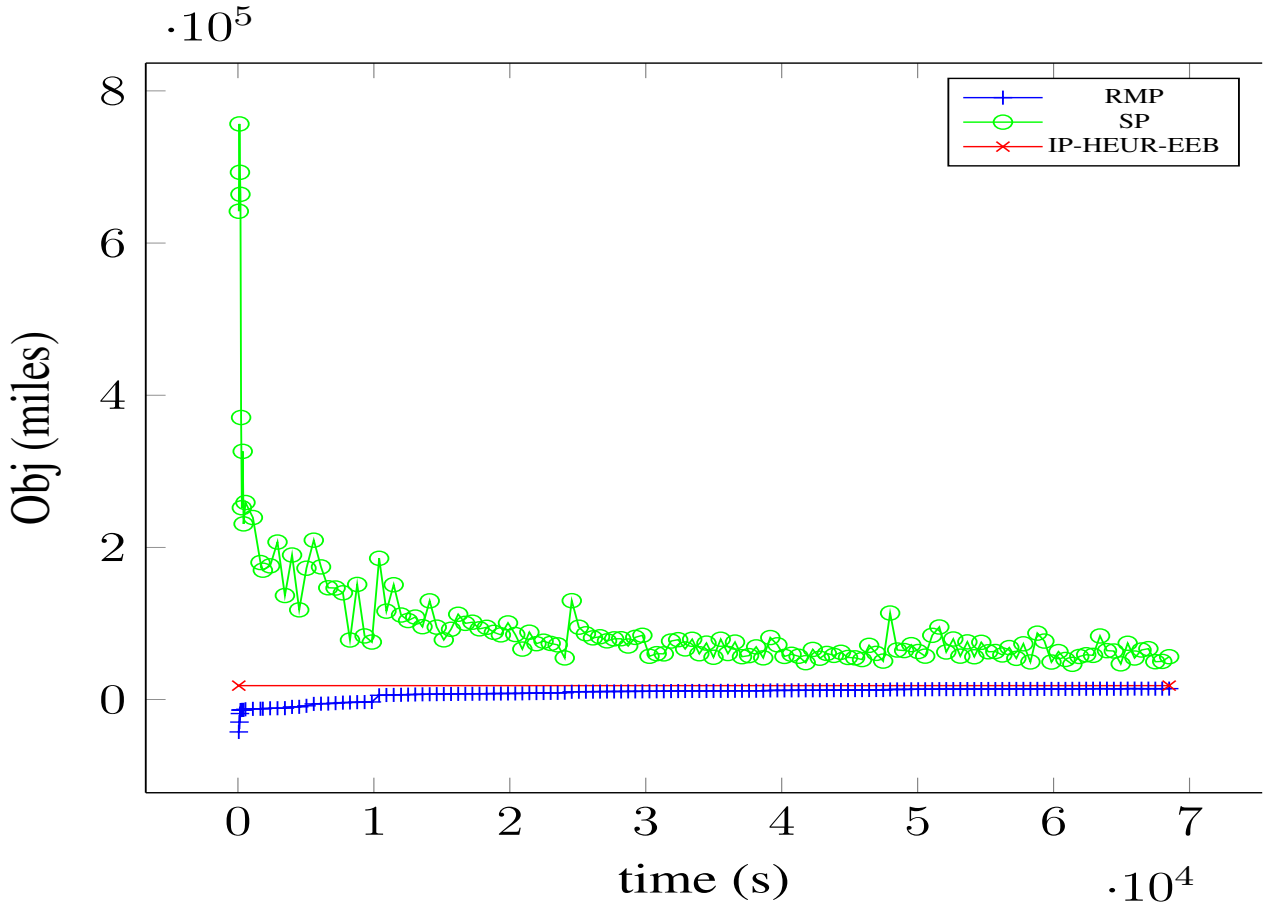


Figure 4.6: Iterations of Benders Decomposition using disaggregated cuts for Instance 14

Using Benders decomposition with aggregated cuts, which we refer to as BD-AGG, also shows a slow rate of convergence; see Figure 4.7 for an example. As expected, using aggregated cuts allows for more iterations within the time limit (336 vs 140 iterations in 5 hours). However, because the aggregated cuts are weaker, this does not result in a better rate of convergence. Table 4.15 summarizes the results of Benders decomposition with aggregated and disaggregated cuts. The results demonstrate that straightforward Benders decomposition implementations perform poorly – SP-Obj provides an upper-bound (as it is associated with a feasible solution) and RMP-Obj provides a lower bound.

To improve the rate of convergence of Benders decomposition, we explore the use of Pareto optimal Benders cuts, as suggested by Magnanti and Wang ([55]). Unfortunately, the auxiliary

Table 4.15: Performance of Benders Decomposition with aggregated and disaggregated cuts compared to IP-HEUR wih E-E-B scheme.

Instance	IP-HEUR			BD-DISAGG			BD-AGG		
	LP-Obj	IP-Obj	TT	RMP-Obj	SP-Obj	TT	RMP-Obj	SP-Obj	TT
11	14,102	14,116	63	2,636	62,621	18,000	-8,023	50,676	18,000
12	11,205	11,207	15	-10,468	66,240	18,000	-14,861	91,582	18,000
13	12,107	12,110	20	7,696	92,942	18,000	-18,366	40,237	18,000
14	18,290	18,300	25	-6,595	86,916	18,000	1,434	113,113	18,000
15	8,069	8,077	18	-20,823	86,012	18,000	-20,839	59,266	18,000

subproblems that have to be solved ($\mathcal{SP}2_e$) exhibit numerical issues. These numerical issues have also been reported by [56] who explain that the issue can arise when the subproblem is solved with column generation, which is the case in our implementation. More specifically, the Magnanti and Wong primal subproblem suffers from numerical unboundedness due to Constraint 4.57. One way to address the issue is to add a small upper bound on variable ξ in $\mathcal{SP}2_e$ (e.g., 0.1). We refer to this approach as BD-MW1. Another way, as proposed by [56], is to eliminating Constraint 4.57 while still using a core point in the subproblem. We refer to this approach as BD-MW2. Finally, there is the approach by [57] which also eliminates Constraint 4.57, but uses a weighted sum of a core point $\hat{y}$ and the optimal solution of the master problem $\bar{y}$ of the form $\bar{y} + \epsilon\hat{y}$ with a sufficiently small ϵ (e.g., 10^{-3}) in the subproblem. We refer to this approach as BD-SL. We assess the performance of these three enhanced approaches using the same instances and with a time limit of 5 hours. The results are summarized in Table 4.16. We observe that BD-MW1 performs best in terms of the rate of convergence for both the relaxed master problem and the subproblem. Figures 4.8 and 4.9 show the convergence profiles of the different approaches for the relaxed master problem and the subproblem, respectively, for Instance 14. We see that all variants of Benders decomposition take many hours to converge to a reasonable gap for 2-day instances. On the other hand, IP-HEUR heuristic with EFFICIENT-ENHANCED-BASIC scheme attains high quality solutions with tight final gaps ($< 0.1\%$) in less than a minute. We conclude that Benders decomposition, even with enhancements, may not be suitable for the short-term inventory-aware

equipment management model as the run-time is excessive.

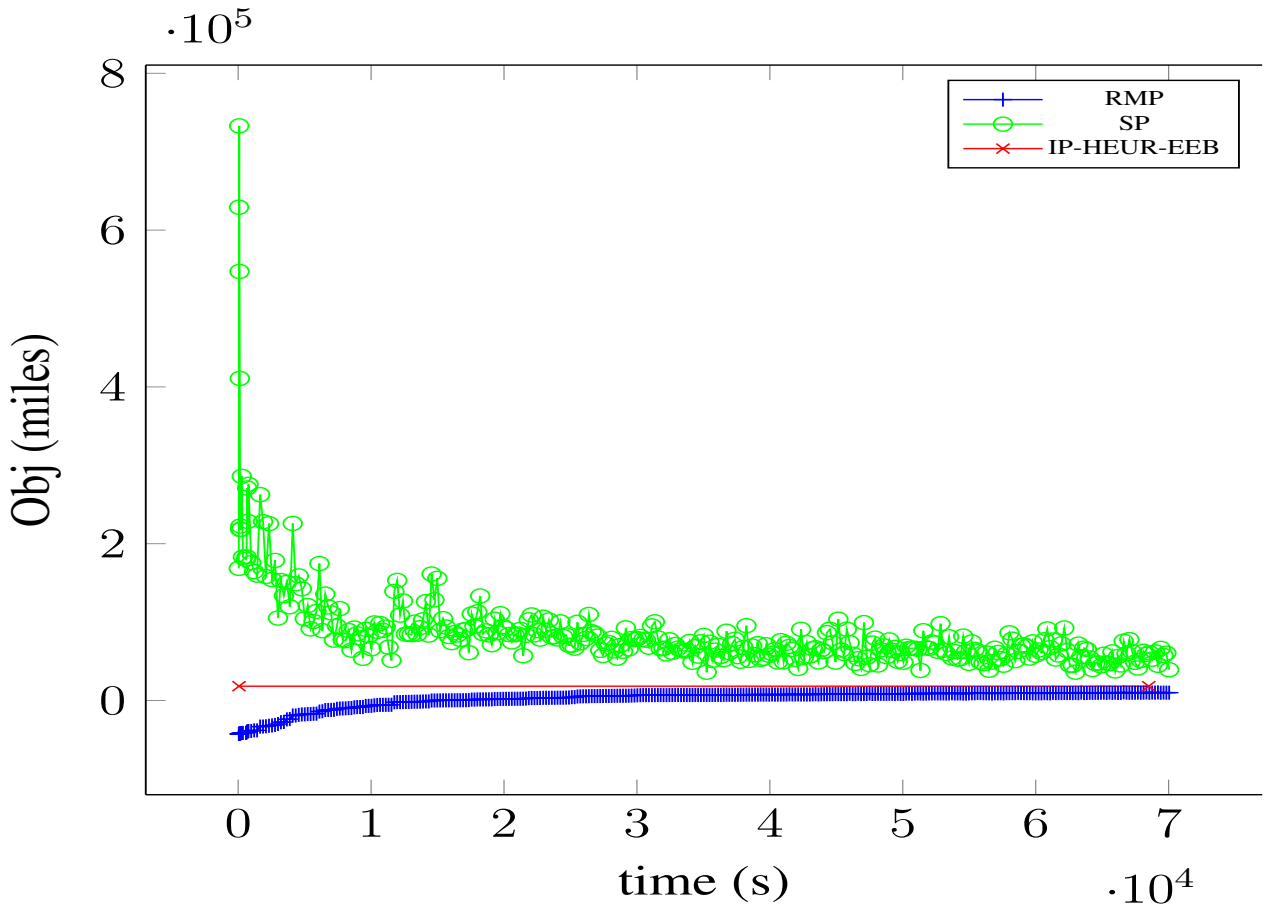


Figure 4.7: Iterations of Benders Decomposition using aggregated Benders cuts for Instance 14

4.7 Final Remarks

We have proposed an inventory-aware equipment management methodology that can be used by logistics companies that operate a heterogeneous fleet of trailers and containers. It relies on substituting equipment types and adding empty repositioning movements. As company networks and fleet sizes can be huge, the methodology uses a parsimonious discretization of time and employs heuristic ideas to efficiently and dynamically generate empty repositioning variables. The methodology produces high quality, but not necessarily optimal, solutions. To obtain optimal solutions,

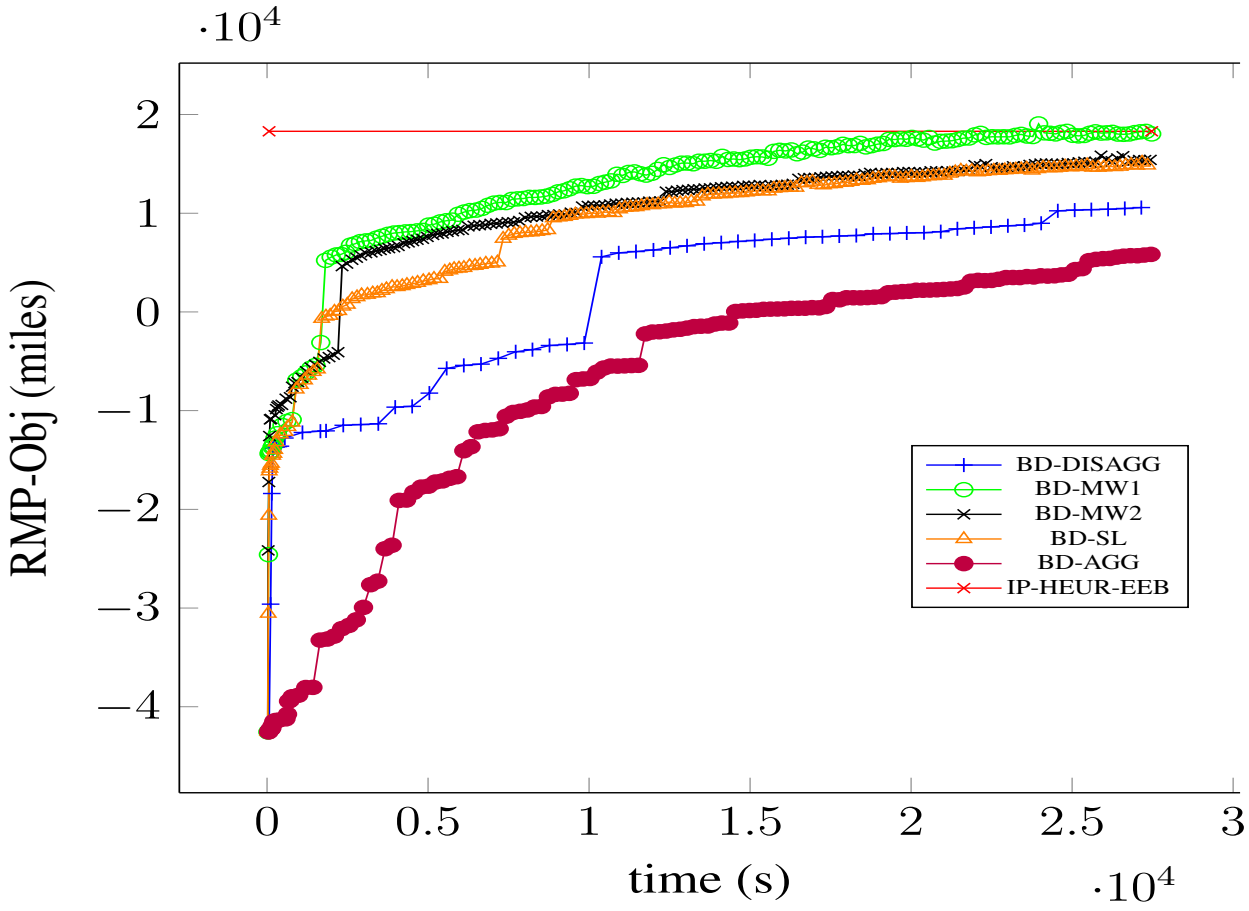


Figure 4.8: Comparison of rate of convergence of the relaxed master problem using different approaches for Instance 14.

techniques such as Branch and Price and Benders decomposition are required but are, at the moment, not practical for real-life instance sizes.

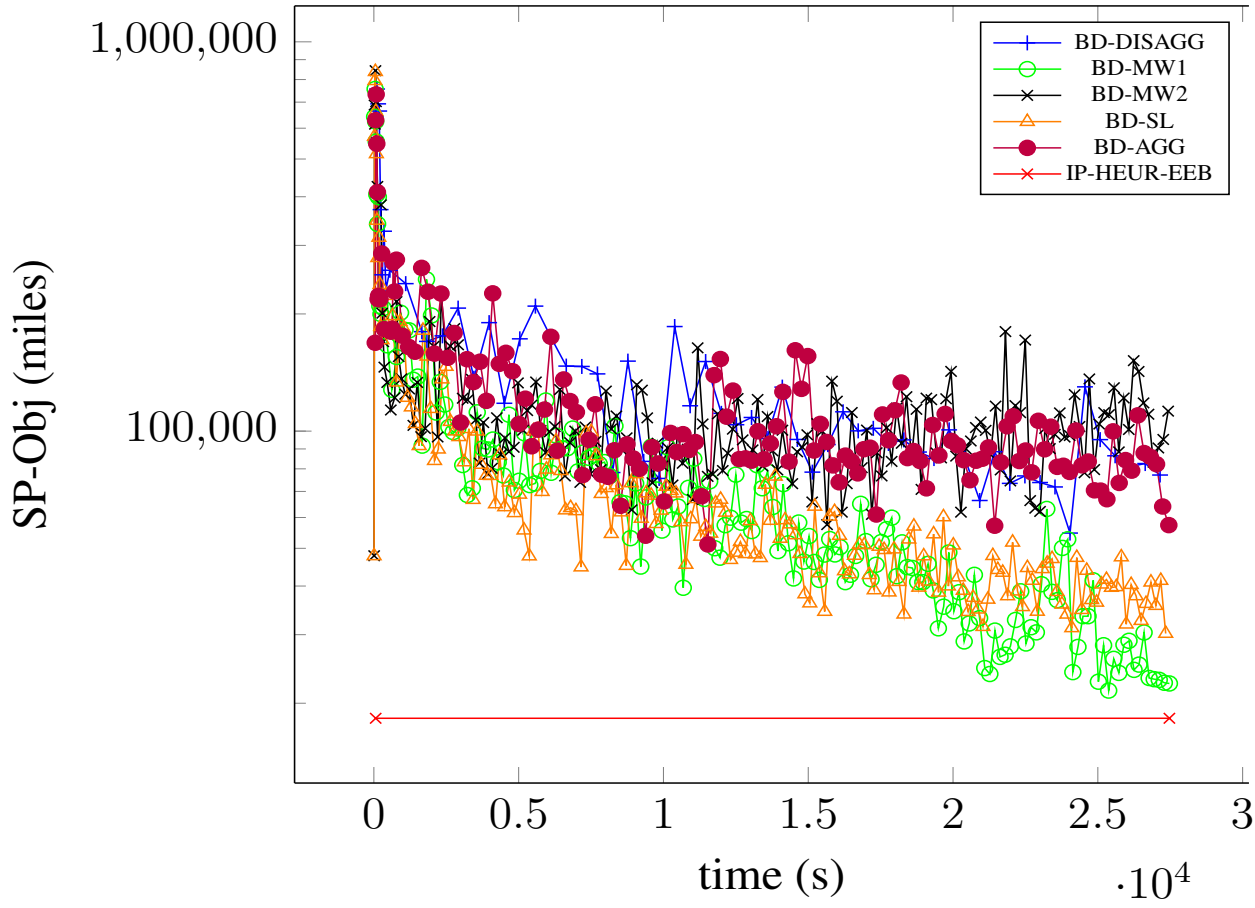


Figure 4.9: Comparison of rate of convergence of the sub-problem using different approaches for Instance 14. We use a logarithmic scale for y-axis to show the difference between the schemes.

Table 4.16: Performance of BD-MW1, BD-MW2, and BD-SL after 5 hours for Instance 14.

Instance	BD-MW1			BD-MW2			BD-SL		
	RMP-Obj	SP-Obj	TT	RMP-Obj	SP-Obj	TT	RMP-Obj	SP-Obj	TT
11	11,597	31,579	18,000	9,382	63,298	18,000	9,599	33,274	18,000
12	7,589	24,727	18,000	655	94,794	18,000	2,580	30,129	18,000
13	8,878	42,041	18,000	6,850	74,541	18,000	4,964	45,448	18,000
14	16,847	42,409	18,000	13,792	83,588	18,000	13,091	38,587	18,000
15	3,351	34,416	18,000	1,556.28	119,149	18,000	-1,425	47,527	18,000

Appendices

APPENDIX A

STATISTICS FOR ALL SHIPMENTS

Tables A.1 and A.2 present statistics for all shipments, i.e., shipments known at time zero and shipments forecast to enter at time 24.

Table A.1: Results for the set of instances I1-I5 used in the computational experiments considering different metrics for all shipments in the system. The best results for each instance in terms of $\boldsymbol{TL}$,$\%\boldsymbol{D}$, $\%\boldsymbol{D}_{\mathbf{OT}}$, and **RO-AVG** are highlighted in bold. $\boldsymbol{TL}$ is in hours, and total runtime **TT** is in seconds.

Ins.	Algorithm	$\boldsymbol{TL}$	$\%\boldsymbol{D}$	$\%\boldsymbol{D}_{\mathbf{OT}}$	$\%\boldsymbol{D}_{\mathbf{L}}$	%PF	%AF	$\%\boldsymbol{S}$	RO-AVG	TT
I1	FIFO-Push	6.00	87.54	63.74	23.81	77.79	21.44	0.72	1.03	535.00
	Urg-Pull	3.72	92.02	73.07	18.95	82.87	16.04	0.59	0.93	424.19
	Urg-Pull-PF	3.65	92.45	72.78	19.66	97.65	0.00	1.14	1.11	170.36
	Blk-LookAhead	3.12	93.30	75.05	18.26	88.13	10.78	0.54	0.70	515.94
	Blk-IP-Basic	3.06	**93.48**	75.40	18.08	91.17	7.69	0.52	0.67	829.54
	Blk-IP-Extended	**3.05**	**93.48**	**75.49**	17.99	91.85	6.98	0.53	**0.57**	8,468.78
I2	FIFO-Push	3.02	92.30	76.72	15.58	68.54	23.80	3.69	0.80	370.37
	Urg-Pull	2.06	94.87	81.96	12.92	71.87	20.45	3.31	0.81	290.67
	Urg-Pull-PF	1.81	95.25	83.08	12.17	88.25	0.00	4.21	0.98	130.72
	Blk-LookAhead	1.39	96.09	85.40	10.70	80.39	11.99	3.33	0.54	366.13
	Blk-IP-Basic	1.33	96.14	85.72	10.42	85.50	5.32	3.28	0.56	550.80
	Blk-IP-Extended	**1.31**	**96.23**	**85.84**	10.38	85.59	5.30	3.31	**0.51**	5,515.14
I3	FIFO-Push	3.56	90.18	74.86	15.32	69.90	23.33	3.83	0.76	363.97
	Urg-Pull	2.31	93.77	80.89	12.88	74.39	19.26	2.74	0.75	323.12
	Urg-Pull-PF	2.04	94.16	81.94	12.22	90.67	0.00	3.51	0.89	140.40
	Blk-LookAhead	1.65	94.90	84.06	10.84	82.22	11.57	2.63	0.45	369.80
	Blk-IP-Basic	1.61	95.10	84.37	10.73	87.40	5.19	2.51	0.47	570.22
	Blk-IP-Extended	**1.60**	**95.13**	**84.49**	10.64	87.51	5.10	2.61	**0.4**	5,842.59
I4	FIFO-Push	5.06	88.91	70.52	18.38	74.45	21.53	3.25	0.70	324.61
	Urg-Pull	3.51	92.56	76.44	16.12	75.82	20.08	3.02	0.73	256.43
	Urg-Pull-PF	3.20	92.84	77.67	15.17	93.32	0.00	4.19	0.88	117.97
	Blk-LookAhead	2.39	94.31	80.28	14.04	84.61	11.39	2.84	0.59	319.58
	Blk-IP-Basic	2.33	94.47	80.68	13.79	89.30	6.38	2.83	0.59	467.32
	Blk-IP-Extended	2.31	94.49	80.70	13.79	89.62	6.05	2.93	0.54	5,531.15
I5	FIFO-Push	2.97	95.43	80.20	15.23	75.98	20.04	1.72	0.91	404.95
	Urg-Pull	2.30	96.41	84.62	11.79	73.69	22.00	1.53	0.91	284.41
	Urg-Pull-PF	1.60	97.53	87.75	9.78	94.04	0.00	1.63	1.06	127.23
	Blk-LookAhead	1.39	97.81	88.87	8.95	84.91	10.91	1.55	**0.75**	368.22
	Blk-IP-Basic	1.33	98.06	**89.28**	8.78	91.22	3.84	1.35	0.78	519.31
	Blk-IP-Extended	**1.31**	**98.09**	89.25	8.84	91.50	3.58	1.42	**0.75**	5,372.42

Table A.2: Results for the set of instances I6-I10 used in the computational experiments considering different metrics for all shipments in the system.. The best results for each instance in terms of $\boldsymbol{TL}$,$\%\boldsymbol{D}$, $\%\boldsymbol{D}_{\mathbf{OT}}$, and **RO-AVG** are highlighted in bold. $\boldsymbol{TL}$ is in hours, and total runtime **TT** is in seconds.

Ins.	Algorithm	$\boldsymbol{TL}$	$\%\boldsymbol{D}$	$\%\boldsymbol{D}_{\mathbf{OT}}$	$\%\boldsymbol{D}_{\mathbf{L}}$	%PF	%AF	$\%\boldsymbol{S}$	RO-AVG	TT
I6	FIFO-Push	2.60	97.59	82.64	14.95	76.63	20.35	1.77	1.12	305.79
	Urg-Pull	2.13	97.97	85.19	12.78	73.11	23.55	1.55	1.04	205.03
	Urg-Pull-PF	1.71	98.73	88.29	10.44	95.33	0.00	1.79	1.21	113.14
	Blk-LookAhead	1.64	98.91	88.53	10.37	84.99	11.87	1.46	0.99	299.84
	Blk-IP-Basic	1.62	98.88	88.65	10.23	93.03	2.81	1.43	1.04	410.41
	Blk-IP-Extended	**1.60**	**98.93**	**88.77**	10.16	93.15	2.99	1.57	**1.01**	3,601.03
I7	FIFO-Push	2.83	95.29	80.58	14.71	76.07	20.00	1.69	0.86	409.99
	Urg-Pull	2.14	96.55	84.62	11.94	73.46	22.11	1.44	0.86	285.85
	Urg-Pull-PF	1.44	97.96	87.74	10.21	94.12	0.00	1.51	1.01	137.23
	Blk-LookAhead	1.18	98.21	89.06	9.16	84.97	10.93	1.32	0.77	373.42
	Blk-IP-Basic	1.16	98.37	89.25	9.12	91.62	3.36	1.29	0.81	530.14
	Blk-IP-Extended	1.15	98.38	89.22	9.16	91.89	3.25	1.32	0.77	5,004.31
I8	FIFO-Push	2.72	95.56	80.76	14.80	76.15	20.15	1.45	0.85	421.18
	Urg-Pull	2.16	96.72	84.57	12.16	73.13	22.59	1.35	0.82	299.62
	Urg-Pull-PF	1.56	97.98	87.77	10.21	94.26	0.00	1.44	0.97	133.76
	Blk-LookAhead	1.43	98.18	88.64	9.53	85.09	11.12	1.09	**0.72**	383.82
	Blk-IP-Basic	1.40	98.26	88.94	9.32	91.69	3.34	1.11	0.76	554.16
	Blk-IP-Extended	**1.37**	**98.31**	**88.99**	9.32	91.86	3.36	1.10	0.73	5,318.20
I9	FIFO-Push	2.66	95.80	81.37	14.43	75.82	20.58	1.60	0.88	416.13
	Urg-Pull	2.09	96.78	84.81	11.96	72.96	22.95	1.44	0.83	290.05
	Urg-Pull-PF	1.57	97.89	87.55	10.34	94.10	0.00	1.79	0.98	127.76
	Blk-LookAhead	1.40	98.22	88.68	9.54	84.41	11.82	1.30	**0.77**	385.93
	Blk-IP-Basic	1.35	**98.25**	**88.83**	9.42	91.70	3.48	1.23	0.80	559.31
	Blk-IP-Extended	**1.34**	**98.25**	88.79	9.46	91.77	3.54	1.41	**0.77**	5,181.67
I10	FIFO-Push	2.95	95.45	78.92	16.53	76.68	20.32	1.71	0.96	424.19
	Urg-Pull	2.37	96.32	82.22	14.10	73.44	22.98	1.74	0.87	297.53
	Urg-Pull-PF	1.93	97.50	85.09	12.41	95.08	0.00	1.84	1.01	137.76
	Blk-LookAhead	1.81	97.76	85.99	11.77	85.28	11.58	1.41	**0.75**	395.01
	Blk-IP-Basic	1.76	97.86	86.27	11.59	92.27	3.57	1.37	0.78	560.56
	Blk-IP-Extended	**1.74**	**97.92**	**86.29**	11.63	92.37	3.51	1.55	**0.75**	5,391.10

APPENDIX B

RESULTS FOR SPATIAL DECOMPOSITION HEURISTIC

We summarize the results of the two variants of spatial decomposition, namely, INTRA-FIRST and INTER-FIRST. We perform three iterations inside these heuristics (i.e., $N_{iter} = 3$).

Table B.1: INTRA-FIRST heuristic results using equipment substitution matrix ESM1.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	3,608	11.96	263	41,124	344	99.16	301
I2	3,982	3,482	12.56	262	41,044	355	99.14	314
I3	4,006	3,488	12.93	262	39,065	370	99.05	306
I4	4,106	3,544	13.69	265	38,069	399	98.95	294
I5	4,126	3,580	13.23	296	38,650	382	99.01	269
I6	3,948	3,448	12.66	275	35,467	361	98.98	316
I7	4,078	3,610	11.48	293	36,501	334	99.08	322
I8	3,870	3,508	9.35	382	36,748	284	99.23	468
I9	3,786	3,454	8.77	298	32,996	258	99.22	332
I10	3,566	3,348	6.11	246	35,078	235	99.33	296

Table B.2: INTRA-FIRST heuristic results using equipment substitution matrix ESM2.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	2,258	44.90	478	36,658	2,337	93.62	757
I2	3,982	2,114	46.91	510	33,327	2,387	92.84	755
I3	4,006	2,096	47.68	467	34,470	2,416	92.99	633
I4	4,106	2,164	47.30	590	33,662	2,433	92.77	706
I5	4,126	2,210	46.44	441	35,451	2,342	93.39	602
I6	3,948	2,152	45.49	454	33,846	2,202	93.49	640
I7	4,078	2,396	41.25	541	35,453	2,130	93.99	796
I8	3,870	2,362	38.97	613	35,487	2,161	93.91	746
I9	3,786	2,310	38.99	504	34,732	2,051	94.09	675
I10	3,566	2,170	39.15	537	36,632	2,128	94.19	710

Table B.3: INTRA-FIRST heuristic results using equipment substitution matrix ESM3.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	1,734	57.69	1,454	40,565	2,520	93.79	2,142
I2	3,982	1,674	57.96	1,373	38,325	2,380	93.79	2,010
I3	4,006	1,706	57.41	1,455	37,293	2,233	94.01	2,179
I4	4,106	1,708	58.40	1,378	37,309	2,215	94.06	2,083
I5	4,126	1,742	57.78	1,081	38,206	2,266	94.07	1,637
I6	3,948	1,674	57.60	1,078	38,157	2,238	94.13	1,684
I7	4,078	1,912	53.11	1,085	34,839	2,189	93.72	1,731
I8	3,870	1,846	52.30	1,110	39,605	2,303	94.19	2,222
I9	3,786	1,804	52.35	1,071	39,685	2,121	94.66	1,600
I10	3,566	1,690	52.61	1,102	40,671	2,251	94.47	1,769

Table B.4: INTER-FIRST heuristic results using equipment substitution matrix ESM2.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	2,260	44.85	674	35,714	2,599	92.72	1,375
I2	3,982	2,124	46.66	584	33,349	2,755	91.74	754
I3	4,006	2,124	46.98	583	34,867	2,650	92.40	1,003
I4	4,106	2,168	47.20	657	34,067	2,661	92.19	735
I5	4,126	2,220	46.19	520	36,818	2,667	92.76	611
I6	3,948	2,142	45.74	531	32,420	2,634	91.88	626
I7	4,078	2,414	40.80	505	33,980	2,430	92.85	677
I8	3,870	2,348	39.33	578	33,325	2,387	92.84	692
I9	3,786	2,308	39.04	561	35,948	2,289	93.63	681
I10	3,566	2,154	39.60	490	34,030	2,275	93.31	857

Table B.5: INTER-FIRST heuristic results using equipment substitution matrix ESM3.

Instance	Phase 1				Phase 2			
	I_0	$\hat{I}$	$\Delta I(\%)$	Time	N_s	$\hat{N}_s$	$\Delta N_s(\%)$	Time
I1	4,098	1,784	56.47	1,088	40,342	2,624	93.50	3,664
I2	3,982	1,778	55.35	1,079	38,270	2,370	93.81	3,014
I3	4,006	1,762	56.02	1,128	37,531	2,543	93.22	2,488
I4	4,106	1,668	59.38	1,312	40,796	2,759	93.24	3,984
I5	4,126	1,772	57.05	1,070	38,948	2,650	93.20	3,201
I6	3,948	1,706	56.79	1,243	36,684	2,429	93.38	2,959
I7	4,078	1,976	51.54	1,237	37,956	2,425	93.61	2,900
I8	3,870	1,866	51.78	1,274	38,472	2,332	93.94	2,486
I9	3,786	1,810	52.19	1,278	38,378	2,229	94.19	2,651
I10	3,566	1,710	52.05	1,341	39,725	2,326	94.14	2,828

Table B.6: A comparison between the exact approach and INTRA-FIRST heuristic for ESM1.

Instance	ESM1					
	Exact			Intra-First		
	$\Delta I(\%)$	$\hat{N}_s$	TT(s)	$\Delta I(\%)$	$\hat{N}_s$	TT(s)
I1	12.15	275	157	11.96	344	564
I2	12.76	268	192	12.56	355	576
I3	13.23	295	274	12.93	370	568
I4	13.74	289	803	13.69	399	559
I5	13.62	316	131	13.23	382	565
I6	13.22	311	237	12.66	361	591
I7	11.72	288	171	11.48	334	615
I8	9.61	222	77	9.35	284	850
I9	8.87	192	97	8.77	258	630
I10	6.23	165	217	6.11	235	542

Table B.7: A comparison between the exact approach, SUB-HEUR, and INTRA-FIRST heuristics for ESM2.

Instance	ESM2								
	Exact			Sub-Heur			Intra-First		
	$\Delta I(\%)$	$\hat{N}_s$	TT(s)	$\Delta I(\%)$	$\hat{N}_s$	TT(s)	$\Delta I(\%)$	$\hat{N}_s$	TT(s)
I1	50.95	2,851	10,014	48.95	2,791	194	44.9	2,337	1,235
I2	52.44	2,817	54,595	50.58	2,845	194	46.91	2,387	1,265
I3	54.19	2,939	10,593	51.55	2,846	324	47.68	2,416	1,100
I4	53.41	2,928	43,883	52.00	2,937	273	47.30	2,433	1,296
I5	52.3	2,885	20,600	50.70	2,887	234	46.44	2,342	1,043
I6	51.47	2,735	13,991	49.92	2,813	405	45.49	2,202	1,094
I7	47.35	2,811	8,574	45.64	2,758	285	41.25	2,130	1,337
I8	46.38	2,933	20,719	44.44	2,866	339	38.97	2,161	1,359
I9	46.22	2,813	11,647	45.06	2,863	312	38.99	2,051	1,179
I10	47.22	2,982	19,530	45.37	2,859	238	39.15	2,128	1,247

Table B.8: A comparison between the exact approach, SUB-HEUR, and INTRA-FIRST heuristics for ESM3.

Instance	ESM3								
	Exact			Sub-Heur			Intra-First		
	$\Delta I(\%)$	$\hat{N}_s$	TT(s)	$\Delta I(\%)$	$\hat{N}_s$	TT(s)	$\Delta I(\%)$	$\hat{N}_s$	TT(s)
I1	68.13	3,394	130,431	62.13	3,423	1,691	57.69	2,520	3,596
I2	69.16	3519	197,582	62.98	3,342	1,475	57.96	2,380	3,383
I3	69.42	3,422	78,562	63.78	3,179	1,556	57.41	2,233	3,634
I4	70.46	3,485	93,075	64.03	3,303	1,610	58.40	2,215	3,461
I5	69.41	3,457	119,000	63.31	14,334	1,864	57.78	2,266	2,718
I6	70.21	3,581	60,099	63.60	3,367	1,677	57.60	2,238	2,762
I7	64.71	3,456	168,382	58.97	13,893	1,467	53.11	2,189	2,816
I8	64.16	3,500	180,522	57.49	3,119	2,523	52.30	2,303	3,332
I9	65.11	3,768	125,226	58.40	3,308	1,558	52.35	2,121	2,671
I10	65.93	3,877	224,418	60.24	14,467	1,725	52.61	2,251	2,871

APPENDIX C

RESTORING BALANCE

As mentioned in the introduction, companies restore balance by introducing empty loads that send equipment from facilities with an excess of equipment to facilities with a deficit of equipment. This incurs extra costs as it may involve creating new driver schedules and additional transportation costs. A natural question to ask is whether it is always possible to restore balance by introducing empty loads. Next, we give a necessary and sufficient condition on the service network that guarantees that balance can be restored by introducing empty loads.

We say a load plan with a single equipment type can be balanced if it is possible to reduce the equipment imbalance to zero by adding empty loads. We first provide an example of a service network that cannot be balanced. Consider network $N = (V, A_0)$ with $V = \{v_1, v_2, v_3, v_4\}$ and $A_0 = \{v_1 \to v_2, v_2 \to v_1, v_2 \to v_3, v_3 \to v_4, v_4 \to v_3\}$ as shown in Figure C.1.

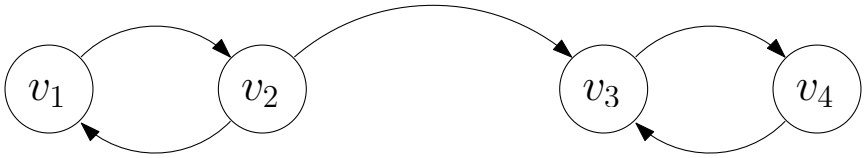
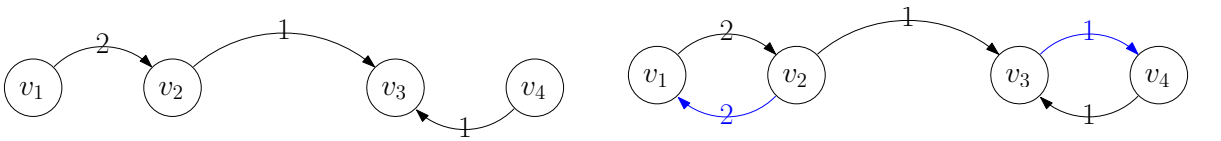


Figure C.1: Example of a network with 4 nodes and 5 arcs.

Let load plan $\mathcal{L}$ have two loads on arc $v_1 \to v_2$, one load on arc $v_2 \to v_3$, and one load on arc $v_4 \to v_3$. Hence, the initial imbalance is 6 (2 at node v_1, 1 at node v_2, 2 at node v_3, and 1 at node v_4). The imbalance can be reduced to 2 by adding two empty loads on the arc $v_2 \to v_1$ and one empty load on arc $v_3 \to v_4$ (see Figure C.2a). It is not possible to reduce the imbalance to zero because there is no path from v_3 to v_2.



(a) Initial load plan $\mathcal{L}$ with imbalance 6

(b) Load plan adjusted with new empty loads (blue arcs) with final imbalance 2

Figure C.2: Example where empty repositioning does not yield zero imbalance

We see that the fact that there is no path from v_3 to v_2 using arcs in A_0 makes it impossible to create a "cycle of loads" to reduce the imbalance. Based on this observation, we give a necessary and sufficient condition that guarantees that a given load plan (with a single equipment type) can be balanced.

Claim 1. *Given a network $N = (V, A_0)$ and a load plan $\mathcal{L}$, then $\mathcal{L}$ can be balanced if and only if every load $\ell \in \mathcal{L}$ belongs to a directed cycle in A_0.*

Proof. Let $|\mathcal{L}(a)|$ be the number of loads on arc $a \in A_0$ in the plan $\mathcal{L}$. The problem of adding empty loads to reach zero imbalance can be viewed as one to find a circulation f such that for any arc $a \in A_0$, the flow satisfies $|\mathcal{L}(a)| \leq f(a)$ and $f(a) \in \mathbb{Z}_+$.

Sufficiency: For any subset $U \subseteq V$, if $\delta^{\text{in}}(U) \neq \emptyset$, then $\delta^{\text{out}}(U) \neq \emptyset$ since each arc belongs to a cycle. Let $d(a) = |\mathcal{L}(a)| \in \mathbb{Z}_+$ and $c(a) = M \in \mathbb{Z}_+$ large enough, $\forall a \in A_0$, then $d(\delta^{\text{in}}(U)) \leq c(\delta^{\text{out}}(U))$. According to Hoffman's circulation theorem ([58]), such circulation f exists.

Necessity: If such a circulation exists, then it can be decomposed into a set of cycles. ■

APPENDIX D

ON THE COMPLEXITY OF THE INTEGRATED MODEL

D.1 Case with full interchangeability

In service networks with a homogeneous fleet equipment management, i.e., empty repositioning of equipment to restore balance, can be modeled as a single-commodity network flow problem and is therefore solvable in polynomial time. When the fleet is heterogeneous, however, equipment management becomes more difficult. This has been shown formally in [44], which presents a complexity analysis of equipment balancing in a flat network with multiple equipment types. The problem becomes NP-hard when the fleet is comprised of three or more equipment types. Not surprisingly, when we allow both equipment substitutions and equipment repositioning, equipment management is also NP-hard. We show that leveraging the known complexity results for Stage 1 of the staged approach provided in [44].

Let $N = (V, A_1)$ be a service network with A_1 the set of loads. Let A_2 represent the set of possible repositioning arcs. We define an *equipement repositioning* (or repositioning for short) to be a function $\mathscr{R} : (A_2, \mathcal{E}) \rightarrow \mathbb{Z}_{\geq 0}$ that assigns a number of empty trailer movements to each arc in A_2 for an equipment type in $\mathcal{E}$. We also define $\|\mathscr{R}\| = \sum_{e \in \mathcal{E}} \sum_{a \in A_2} \mathscr{R}(a, e) * D_{ae}$ the total cost of a repositioning. As defined in Chapter 3, an equipment assignment is a function $\mathscr{A} : A \rightarrow \mathcal{C}$ that assigns an equipment type to each arc. The complexity of integrated equipment management is established in the following proposition.

Proposition 3. *The problem of finding whether there exists an assignment $\mathscr{A}$ and a repositioning $\mathscr{R}$ that restores balance such that $\|\mathscr{A} - \mathscr{A}_0\| \leq K_1$ and $\|\mathscr{R}\| = K_2$, for a given $K_1, K_2 \in \mathbb{Z}_{\geq 0}$ and a network N with three equipment types and full interchangeability is NP-complete.*

Proof. Transformation from 3FI PROBLEM (3-equipment with full interchangeability) – defined

and shown to be NP-complete in [44].

3FI PROBLEM: Given a network N with three equipment types and full interchangeability and an integer $K \in \mathbb{Z}_{\geq 0}$, does there exist an assignment $\mathscr{A}$ such that $I(\mathscr{A}) = I^*$ and $\|\mathscr{A} - \mathscr{A}_0\| \leq K$?

We create one instance of the integrated model from the same network N. We assume $K_1 = K$ and $K_2 = \frac{I^*}{2}$ where $I(\mathscr{A}) = I^*$. We also assume that the repositioning cost D_{ae} is constant and is equal to 1 for all repositioning arcs and equipment types:

$$D_{ae} = 1, \quad \forall a \in A_2, e \in \mathcal{E}.$$

We start from the observation that a repositioning movement reduces the imbalance in the network by two units, one at the origin and one at the destination. Given that the combined assignment $\mathscr{A}$ and repositioning $\mathscr{R}$ restore balance in the network, we have that:

$$\sum_{e \in \mathcal{E}} \sum_{a \in A_2} \mathscr{R}(a, e) = \frac{I(\mathscr{A})}{2}.$$

Given that the repositioning cost is constant and is equal to one, this yields that:

$$\|\mathscr{R}\| = \frac{I(\mathscr{A})}{2}$$

With this result, a YES-instance of the 3FI PROBLEM gives a YES-instance of the integrated model as $\|\mathscr{R}\| = \frac{I^*}{2}$ and $\|\mathscr{A} - \mathscr{A}_0\| \leq K$. Conversely, a YES-instance of the integrated model that satisfies $\|\mathscr{R}\| = \frac{I^*}{2}$ and $\|\mathscr{A} - \mathscr{A}_0\| \leq K$, gives a YES-instance of the 3FI PROBLEM. ■

A Polynomially solvable case: Phase 1 with full interchangeability

Here, we consider the case where there is full interchangeability between equipment types and we assume the repositioning cost $D_{ae} = D_a$ only depends on the arc and not the equipment type. The

second assumption is reasonable for the major logistics companies as the cost depends more on the mileage and the lane and not the type of the equipment. We can show that Phase 1 of the integrated model is polynomially solvable. It suffices to assign the same equipment type to all the loads then solve a single commodity network flow problem to restore the remaining imbalance with the least repositioning cost. To prove this claim we use the following proposition.

Proposition 4. *In a network N with full interchangeability between equipment types, from any optimal solution of Phase 1 of the integrated model, it is possible to construct another optimal solution where only one equipment type is used in both the assignment and repositioning.*

Proof. As all equipment types are interchangeable, we randomly pick one specific equipment type e^*. In the optimal solution, we substitute the equipment type of all the loads to e^*. In the repositioning solution, we also assign e^* to all the repositioning movements generated in the optimal solution. We can prove that this new solution is feasible and optimal. For feasibility, it suffices to prove that it restores balance in all the facilities. At a given facility i, for a given equipment type $e' \neq e^*$, given that all the loads and repositioning movements that use this equipment type e' are such that the imbalance is zero in the optimal solution, substituting all of them to e^* does not impact the imbalance of e^* as we add the same number of inbound and outbound arcs with equipment type e^* to i. Any equipment type other than e^* still maintains a zero imbalance in i as there are no longer loads or repositioning movements carrying this equipment type. This shows that the new solution maintains zero imbalance. For optimality, as the repositioning cost does not depend on the equipment type and since we are keeping the same repositioning arcs in the optimal solution and only change the equipment type assigned to them to be e^*, the total repositioning cost $\|\mathscr{R}\|$ does not change and remains optimal. ■

Using Proposition 4, we can fix the variables y in Phase 1 model to $y_{le^*} = 1$ and still guarantee that we achieve the optimal repositioning cost. This results in a single commodity network flow problem as we are only balancing e^*. This proves the polynomial solvability of this special case.

Note that Phase 2 remains a priori NP-hard even with full interchangeability based on Proposition 3.

D.2 Case with partial interchangeability

When there is partial interchangeability between equipment types, the complexity Stage 1 of the staged approach increases as shown in [44]. For the integrated model, we can also show that Phase 1 is no longer polynomially solvable. For that, we will need an intermediate result that extends a complexity result from [44]. [44] proved that: *the problem of deciding whether there exists an assignment $\mathscr{A}$ such that $I(\mathscr{A}) = 0$ for a balanced network N with three equipment types and a set $S \subseteq A$ of arcs on which the equipment type cannot be changed is NP-complete.* We can extend this result to general non balanced networks through the following proposition.

Proposition 5. *The problem of deciding whether there exists an assignment $\mathscr{A}$ such that $I(\mathscr{A}) = I^*$ for a network N with three equipment types and a set $S \subseteq A$ of arcs on which the equipment type cannot be changed is NP-complete.*

Proof. Transformation from 3PI PROBLEM. (3-equipment with partial interchangeability), defined and shown to be NP-complete in [44].

3PI PROBLEM: Given a balanced network N with three equipment types and a set $S \subseteq A$ of arcs on which the equipment type cannot be changed, does there exist an assignment $\mathscr{A}$ such that $I(\mathscr{A}) = 0$?

We will refer to our problem as 3NPI (3-equipment non balanced network with partial interchangeability). Given a balanced network $N = (V, A)$ with three equipment types and a set $S \subseteq A$ of arcs on which the equipment type cannot be changed, we create a new network $N' = (V, A')$ as follows. We add a new equipment e' to $\mathcal{E}$ to form a new set of equipment types $\mathcal{E}'$. We add one new arc a' with this equipment between two existing nodes in N to form a new network $N' = (V, A')$

where $A' = A \cup \{a'\}$. We assume the equipment type e' cannot be changed on the arc a'. we also define $S' = A \cup \{a'\}$ as the set of arcs on which equipment type cannot be changed in N'. Since N was assumed to be balanced, then N' will have a minimal imbalance of 2 due to the new arc a', i.e., $I^* = 2$. It is trivial to see that a YES-instance of the 3PI problem yields a YES-instance of 3NPI problem and vice-versa. ■

We use Proposition 5 to show the complexity of solving Phase 1 of the integrated model when there is partial interchangeability between equipment types, through the following proposition.

Proposition 6. *The problem of finding whether there exists an assignment $\mathscr{A}$ and a repositioning $\mathscr{R}$ that restores balance such that $\|\mathscr{R}\| = K$, for a given $K \in \mathbb{Z}_{\geq 0}$ and a network N with three equipment types and a set $S \subseteq A$ of arcs on which the equipment type cannot be changed is NP-complete.*

Proof. Transformation from 3NPI PROBLEM. The proof proceeds analogous to the proof of Proposition 3. We assume $K = \frac{I^*}{2}$. . ■

REFERENCES

[1] ATRI, *E-commerce impacts on the trucking industry*, https://truckingresearch.org/wp-content/uploads/2019/02/ATRI-Impacts-of-E-Commerce-on-Trucking-02-2019.pdf.

[2] R. A. Garrido and H. S. Mahmassani, "Forecasting freight transportation demand with the space–time multinomial probit model," *Transportation Research Part B: Methodological*, vol. 34, no. 5, pp. 403–418, 2000.

[3] C. Winston, "The demand for freight transportation: Models and applications," *Transportation Research Part A: General*, vol. 17, no. 6, pp. 419–427, 1983.

[4] A. Nuzzolo and A. Comi, "Urban freight demand forecasting: A mixed quantity/delivery/vehicle-based model," *Transportation Research Part E: Logistics and Transportation Review*, vol. 65, pp. 84–98, 2014.

[5] T. G. Crainic, "Service network design in freight transportation," *European Journal of Operational Research*, vol. 122, no. 2, pp. 272–288, 2000.

[6] N. Wieberneit, "Service network design for freight transportation: A review," *OR spectrum*, vol. 30, no. 1, pp. 77–112, 2008.

[7] A. Baubaid, N. Boland, and M. Savelsbergh, "The value of limited flexibility in service network designs," Optimization Online 7102, 2019.

[8] A. L. Erera, M. Hewitt, M. W. Savelsbergh, and Y. Zhang, "Creating schedules and computing operating costs for ltl load plans," *Computers & Operations Research*, vol. 40, no. 3, pp. 691–702, 2013.

[9] T. G. Crainic and G. Laporte, "Planning models for freight transportation," *European journal of operational research*, vol. 97, no. 3, pp. 409–438, 1997.

[10] J. M. Farvolden and W. B. Powell, "Subgradient methods for the service network design problem," *Transportation Science*, vol. 28, no. 3, pp. 256–272, 1994.

[11] L. Barcos, V Rodríguez, M. J. Álvarez, and F. Robusté, "Routing design for less-than-truckload motor carriers using ant colony optimization," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 3, pp. 367–383, 2010.

[12] T. G. Crainic, M. Gendreau, and J. M. Farvolden, "A simplex-based tabu search method for capacitated network design," *INFORMS journal on Computing*, vol. 12, no. 3, pp. 223–236, 2000.

[13] I. Ghamlouche, T. G. Crainic, and M. Gendreau, "Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design," *Annals of Operations research*, vol. 131, no. 1-4, pp. 109–133, 2004.

[14] A. Erera, M. Hewitt, M. Savelsbergh, and Y. Zhang, "Improved load plan design through integer programming based local search," *Transportation Science*, vol. 47, no. 3, pp. 412–427, 2013.

[15] K. Holmberg and D. Yuan, "A lagrangean approach to network design problems," *International Transactions in Operational Research*, vol. 5, no. 6, pp. 529–539, 1998.

[16] N. Katayama and S. Yurimoto, "The load planning problem for less-than-truckload motor: Carriers and a solution approach," in *Developments in Logistics and Supply Chain Management*, Springer, 2016, pp. 240–249.

[17] A. I. Jarrah, E. Johnson, and L. C. Neubert, "Large-scale, less-than-truckload service network design," *Operations Research*, vol. 57, no. 3, pp. 609–625, 2009.

[18] M Gendreau, P Badeau, F Guertin, J. Potvin, and E Taillard, "A solution procedure for real-time routing and dispatching of commercial vehicles," in *Intelligent Transportation: Realizing the Future. Abstracts of the Third World Congress on Intelligent Transport SystemsITS America*, 1996.

[19] A. Regan, H. Mahmassani, and P. Jaillet, "Dynamic decision making for commercial fleet operations using real-time information," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1537, pp. 91–97, 1996.

[20] J. Yang, P. Jaillet, and H. Mahmassani, "On-line algorithms for truck fleet assignment and scheduling under real-time information," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1667, pp. 107–113, 1999.

[21] R. K. Cheung and B Muralidharan, "Impact of dynamic decision making on hub-and-spoke freight transportationnetworks," *Annals of Operations Research*, vol. 87, pp. 49–71, 1999.

[22] ——, "Dynamic routing for priority shipments in ltl service networks," *Transportation science*, vol. 34, no. 1, pp. 86–98, 2000.

[23] J. Roy, "Recent trends in logistics and the need for real-time decision tools in the trucking industry," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, IEEE, 2001.

[24] ——, "The impact of new supply chain management practices on the decision tools required by the trucking industry," in *Applications of Supply Chain Management and E-Commerce Research*, Springer, 2005, pp. 119–140.

[25] W. B. Powell, "Dynamic models of transportation operations," *Handbooks in Operations Research and management science*, vol. 11, pp. 677–756, 2003.

[26] B. Hejazi and A. Haghani, "Dynamic decision making for less-than-truckload trucking operations," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2032, pp. 17–25, 2007.

[27] W. B. Powell, A. Marar, J. Gelfand, and S. Bowers, "Implementing real-time optimization models: A case application from the motor carrier industry," *Operations Research*, vol. 50, no. 4, pp. 571–581, 2002.

[28] T. G. Crainic, M. Gendreau, and J.-Y. Potvin, "Intelligent freight-transportation systems: Assessment and the contribution of operations research," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 541–557, 2009.

[29] A.-K. Rothenbächer, M. Drexl, and S. Irnich, "Branch-and-price-and-cut for a service network design and hub location problem," *European Journal of Operational Research*, vol. 255, no. 3, pp. 935–947, 2016.

[30] C. Barnhart, N. Krishnan, D. Kim, and K. Ware, "Network design for express shipment delivery," *Computational Optimization and Applications*, vol. 21, no. 3, pp. 239–262, 2002.

[31] H. Simao and W. Powell, "Decomposition Methods for Dynamic Load Planning and Driver Management in LTL Trucking," in *Odysseus*, *Workshop on Freight Transportation and Logistics, Odysseus*, 2018.

[32] Y. Du and R. Hall, "Fleet sizing and empty equipment redistribution for center-terminal transportation networks," *Management Science*, vol. 43, no. 2, pp. 145–157, 1997.

[33] M. Boile, S. Theofanis, A. Baveja, and N. Mittal, "Regional repositioning of empty containers: Case for inland depots," *Transportation Research Record*, vol. 2066, no. 1, pp. 31–40, 2008.

[34] A. L. Erera, J. C. Morales, and M. Savelsbergh, "Robust optimization for empty repositioning problems," *Operations Research*, vol. 57, no. 2, pp. 468–483, 2009.

[35] Y. Long, L. H. Lee, and E. P. Chew, "The sample average approximation method for empty container repositioning with uncertainties," *European Journal of Operational Research*, vol. 222, no. 1, pp. 65–75, 2012.

[36] P. J. Dejax and T. G. Crainic, "Survey paper—a review of empty flows and fleet management models in freight transportation," *Transportation science*, vol. 21, no. 4, pp. 227–248, 1987.

[37] H. Chang, H. Jula, A. Chassiakos, and P. Ioannou, "A heuristic solution for the empty container substitution problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 2, pp. 203–216, 2008.

[38] A. Baykasoğlu, K. Subulan, A. S. Taşan, and N. Dudaklı, "A review of fleet planning problems in single and multimodal transportation systems," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 631–697, 2019.

[39] J. A. Carbajal, A. Erera, and M. Savelsbergh, "Balancing fleet size and repositioning costs in ltl trucking," *Annals of Operations Research*, vol. 203, no. 1, pp. 235–254, 2013.

[40] T. G. Crainic, M. Gendreau, and P. Dejax, "Dynamic and stochastic models for the allocation of empty containers," *Operations research*, vol. 41, no. 1, pp. 102–126, 1993.

[41] A. Imai and F. Rivera IV, "Strategic fleet size planning for maritime refrigerated containers," *Maritime Policy & Management*, vol. 28, no. 4, pp. 361–374, 2001.

[42] O. Jabali, M. Gendreau, and G. Laporte, "A continuous approximation model for the fleet composition problem," *Transportation Research Part B: Methodological*, vol. 46, no. 10, pp. 1591–1606, 2012.

[43] J Gould, "The size and composition of a road transport fleet," *Journal of the Operational Research Society*, vol. 20, no. 1, pp. 81–92, 1969.

[44] Y Yang, Y Ridouane, N Boland, A Erera, and M Savelsbergh, "Substitution-based equipment balancing in service networks with multiple equipment types,"

[45] R. A. Rushmeier and S. A. Kontogiorgis, "Advances in the optimization of airline fleet assignment," *Transportation science*, vol. 31, no. 2, pp. 159–169, 1997.

[46] B. B. Oliveira, M. A. Carravilla, and J. F. Oliveira, "Fleet and revenue management in car rental companies: A literature review and an integrated conceptual framework," *Omega*, vol. 71, pp. 11–26, 2017.

[47] B. D. Brouer, D. Pisinger, and S. Spoorendonk, "Liner shipping cargo allocation with repositioning of empty containers," *INFOR: Information Systems and Operational Research*, vol. 49, no. 2, pp. 109–124, 2011.

[48] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting stock problem—part ii," *Operations research*, vol. 11, no. 6, pp. 863–888, 1963.

[49] A. A. Farley, "A note on bounding a class of linear programming problems, including cutting stock problems," *Operations Research*, vol. 38, no. 5, pp. 922–923, 1990.

[50] L. Gouveia, "Multicommodity flow models for spanning trees with hop constraints," *European Journal of Operational Research*, vol. 95, no. 1, pp. 178–190, 1996.

[51] D.-S. Choi and I.-C. Choi, "On the effectiveness of the linear programming relaxation of the 0-1 multi-commodity minimum cost network flow problem," in *International Computing and Combinatorics Conference*, Springer, 2006, pp. 517–526.

[52] P. Cappanera and G. Gallo, "A multicommodity flow approach to the crew rostering problem," *Operations Research*, vol. 52, no. 4, pp. 583–596, 2004.

[53] M. Moz and M. V. Pato, "An integer multicommodity flow model applied to the rerostering of nurse schedules," *Annals of Operations Research*, vol. 119, no. 1-4, pp. 285–301, 2003.

[54] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Computational Management Science*, vol. 2, no. 1, pp. 3–19, 2005.

[55] T. L. Magnanti and R. T. Wong, "Accelerating benders decomposition: Algorithmic enhancement and model selection criteria," *Operations research*, vol. 29, no. 3, pp. 464–484, 1981.

[56] N. Papadakos, "Practical enhancements to the magnanti–wong method," *Operations Research Letters*, vol. 36, no. 4, pp. 444–449, 2008.

[57] H. D. Sherali and B. J. Lunday, "On generating maximal nondominated benders cuts," *Annals of Operations Research*, vol. 210, no. 1, pp. 57–72, 2013.

[58] A. J. Hoffman, "Some recent applications of the theory of linear inequalities to extremal combinatorial analysis," in *Selected Papers Of Alan J Hoffman: With Commentary*, World Scientific, 2003, pp. 244–258.

VITA

Yassine Ridouane was born in Morocco in the village of Ait Aissa Ouaali located in the Dades Valley. He belongs to the Ait Atta of Sahara, a large Amazigh tribal confederation based in the southern part of the High Atlas Mountains of Morocco.

After accomplishing his secondary school studies majoring in Mathematics, Yassine moved to France to pursue Engineering studies. He pursued the *CPGE* program in Physics and Engineering Sciences at Lycee Faidherbe. After taking the national entrance exam to engineering schools, he was admitted to Ecole Centrale Paris where he received a Bachelor's and a Master's degree of Science in Energy Systems and Process Engineering. Yassine received a Merit-based scholarship from the Moroccan government upon his admission at Ecole Centrale Paris. After finishing his second year, he took a gap year and worked as an Algorithmics intern at Leap Energy Technology Ventures, an Oil and Gas consulting company based in Malaysia and Australia, where he worked on developing Optimization and Simulation based tools to assist Oil and Gas firms in their upstream operations. In summer 2014, Yassine received the Fulbright Scholarship and moved to the United States to pursue a Master's degree in Operations Research at the Georgia Institute of Technology. After accomplishing his degree, he was admitted to the PhD program in the same department where he pursued a Doctorate degree in Industrial Engineering with a concentration in Supply Chain Engineering. His research focus is in the design of efficient algorithmic approaches to solve large scale problems in the realm of Transportation Science and Logistics.

Yassine joined Amazon.com as a Research Scientist in summer 2020. He is working on solving transportation and logistics problems within the Middle Mile team.